Samuel Ady Sanjaya

Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms

Quick Submit
Quick Submit
Universitas Multimedia Nusantara

Document Details

Submission ID
trn:oid:::1:3308505772

Submission Date
Aug 5, 2025, 11:52 AM GMT+7

Download Date
Aug 5, 2025, 11:58 AM GMT+7

File Name
Adverse_Media_Classification_Document_Combined.pdf

File Size
30.2 MB

196 Pages

135,206 Words

743,041 Characters

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

540 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
5% Publications
3% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **540** Not Cited or Quoted 7%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
5% Publications
3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
Technological Institute of the Philippines		2%
2	Internet	
library.cbit.ac.in		<1%
3	Internet	
hk.aconf.org		<1%
4	Publication	
"Proceedings of the 2nd International Conference on Big Data, IoT and Machine L...		<1%
5	Publication	
Sakorn Mekruksavanich, Ponnipa Jantawong, Anuchit Jitpattanakul. "A Hybrid De...		<1%
6	Internet	
hyokadb02.jimu.kyutech.ac.jp		<1%
7	Publication	
Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...		<1%
8	Publication	
"Data Science and Applications", Springer Science and Business Media LLC, 2025		<1%
9	Publication	
R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P...		<1%
10	Publication	
"Advances in Artificial-Business Analytics and Quantum Machine Learning", Sprin...		<1%

11	Publication	"ECAI 2020", IOS Press, 2020 <1%
12	Publication	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng... <1%
13	Publication	"Proceedings of Sixth International Conference on Computer and Communicatio... <1%
14	Publication	"Innovations in Electrical and Electronics Engineering", Springer Science and Busi... <1%
15	Internet	www.mdpi.com <1%
16	Publication	"Advances in Computational Intelligence", Springer Science and Business Media L... <1%
17	Publication	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in He... <1%
18	Publication	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Co... <1%
19	Publication	"Advances in Artificial Intelligence, Big Data and Algorithms", IOS Press, 2023 <1%
20	Publication	"Proceedings of the International Conference on Artificial Intelligence and Comp... <1%
21	Publication	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artific... <1%
22	Publication	"Proceedings of International Conference on Recent Trends in Computing", Sprin... <1%
23	Publication	"Data Science and Big Data Analytics", Springer Science and Business Media LLC, ... <1%
24	Publication	"Soft Computing for Security Applications", Springer Science and Business Media ... <1%

25 Publication
"Proceedings of the 12th International Conference on Soft Computing and Patter... <1%

26 Publication
Oluwatobi Adeleke, Sina Karimzadeh, Tien-Chien Jen. "Machine Learning-Based M... <1%

27 Publication
Sakorn Mekruksavanich, Ponnipa Jantawong, Narit Hnoohom, Anuchit Jitpattana... <1%

28 Publication
"International Conference on Innovative Computing and Communications", Sprin... <1%

29 Publication
"Computational Intelligence for Engineering and Management Applications", Spri... <1%

30 Publication
Salwa Belaqziz, Salma El Hajjami, Hicham Amellal, Redouan Lahmyed, Lahcen Kou... <1%

31 Publication
"Advances in Artificial Intelligence and Data Engineering", Springer Science and B... <1%

32 Student papers
National College of Ireland <1%

33 Student papers
Coventry University <1%

34 Student papers
Liverpool John Moores University <1%

35 Publication
Sajjan Singh, Sarabpreet Kaur. "Latest Trends in Engineering and Technology - AI... <1%

36 Publication
"Second International Conference on Image Processing and Capsule Networks", S... <1%

37 Publication
"PRICAI 2018: Trends in Artificial Intelligence", Springer Science and Business Me... <1%

38 Publication
zhonglin zhao, Baohong Lu, Shuo Zhang, Daoli Wang, Jiaquan Wan, Ranyu Liu, Hu... <1%

39 Publication

Amit Kumar Tyagi, Ajith Abraham. "Recurrent Neural Networks", CRC Press, 2022 <1%

40 Publication

Harahsheh, Khawlah. "Enhancing IoT Security Using Lightweight Machine Learni... <1%

41 Publication

Sharafkhani, Fahimeh. "Deep Learning and Adaptive Clustering Approaches for Fl... <1%

42 Publication

"Computational Collective Intelligence", Springer Science and Business Media LLC... <1%

43 Publication

He Rao, Hongfei Zhan, Rui Wang, Junhe Yu. "A Lightweight and Enhanced YOLO11... <1%

44 Internet

jtsiskom.undip.ac.id <1%

45 Publication

"Cyber Security and Computer Science", Springer Science and Business Media LLC... <1%

46 Publication

Bidong Chen, Lingui Li, Yuanda Lin, Xu yang, Sio Kei Im, RuiPedro Paiva, Yapeng ... <1%

47 Publication

Inam Ullah Khan, Salma El Hajjami, Mariya Ouaissa, Salwa Belaqziz, Tarandeep Ka... <1%

48 Publication

Sunilkumar Dube, Dube Swaraj. "Edge Computing Techniques for Classification b... <1%

49 Publication

"Innovative Data Communication Technologies and Application", Springer Scienc... <1%

50 Publication

Mohamed Elbleihy, Dorota Wolak, Amir Khan, Aneel Manan, Kennedy C. Onyelow... <1%

51 Publication

Shai Tishby Tamari, Yoav Rubinstein, Netta Livneh, Maayan Moshkovitz, Abeer Ka... <1%

52 Publication

Song Gao, Yingjie Hu, Wenwen Li. "Handbook of Geospatial Artificial Intelligence"... <1%

53	Publication	Ahmed A. Abd El-Latif, Mohammed A ElAffendi, Mohamed Ali AlShara, Yassine Ma...	<1%
54	Publication	Deema Mohammed AlSekait, Mohammed Zakariah, Syed Umar Amin, Zafar Iqbal ...	<1%
55	Student papers	Ravensbourne	<1%
56	Publication	Sujata Dash, Subhendu Kumar Pani, Joel J. P. C. Rodrigues, Babita Majhi. "Deep Le...	<1%
57	Student papers	University of Melbourne	<1%
58	Publication	Thiago Henrique Segreto Silva. "High-performance leaf segmentation and detecti...	<1%
59	Publication	d'Oliveira, Pedro Afonso Marques. "Navigating the Mobile App Galaxy: Harnessin...	<1%
60	Publication	Amit Kumar Tyagi. "Data Science and Data Analytics - Opportunities and Challeng...	<1%
61	Publication	H L Gururaj, M R Pooja, Francesco Flammini. "Recent Trends in Computational Sci...	<1%
62	Publication	Maggie Lieu, Qi Cao, Ruoqing Meng, Dong Zhang, Veeren M. Chauhan. "WormAI: ...	<1%
63	Publication	Islam, Tunazzina. "Understanding and Analyzing Microtargeting Pattern on Socia...	<1%
64	Publication	Stylianos Mavrianos, Vera van Santvoort, Sven Teurlincx, Steven AJ Declerk, Kathr...	<1%
65	Internet	www.manit.ac.in	<1%
66	Internet	wwmms.up.ac.th	<1%

67 Publication

Samarjeet Borah, Ratna Raja Kumar Jambi, Sharifah Sakinah Syed Ahmad, Mahen... <1%

68 Publication

Alireza Ghotbi, Mohammad Rahimi, Ahmad Zamani. "Performance Evaluation of L... <1%

69 Publication

Arvind Mewada, Nagendra Singh, Mohd Aquib Ansari, Amrendra Singh Yadav. "A... <1%

70 Publication

Canedo, Daniel Duarte. "Data-Centric Artificial Intelligence in the Context of Com... <1%

71 Publication

Celorico, Maria Inês Monteiro. "Classic Car Spare Parts for Restoration: Internatio... <1%

72 Publication

Tian, Yuan. "Inferring user Needs & Tasks from App Usage Interactions", Universi... <1%

73 Publication

Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "Internationa... <1%

74 Publication

Gonçalves, Filipe Manuel Carvalho Rodrigues Bravo. "Human Well-Being Monitori... <1%

75 Publication

Hamad, Rebeen Ali. "Sequential Learning and Shared Representation for Sensor-... <1%

76 Publication

Manisha Guduri, V Uma Maheswari, Rajanikanth Aluvalu, Amit Krishna Dwivedi, ... <1%

77 Publication

Tiago Augusto Orcajo Demay Cordeiro. "Detecção de anomalias na comunicação ... <1%

78 Publication

Vandana Mohindru Sood, Yashwant Singh, Bharat Bhargava, Sushil Kumar Naran... <1%

turnitin Page 9 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms

1st Reza Juliandri
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
reza.juliandri@student.umn.ac.id

2nd Monika Evelin Johan
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

3rd Jansen Wiratama
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

4th Samuel Ady Sanjaya
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

***Abstract*— Adverse media is negative information that is not profitable for businesses or individuals, while adverse media classification is the process of classifying news titles that are included in adverse media. In an effort to create a system capable of mitigating the occurrence of fraud for customer satisfaction, machine learning is used to classify news both as detrimental media and not for the selection of news for the customer due diligence system. This study utilizes the XGBoost and Gradient Boosting algorithms to classify news headlines. A data set of 1,281 records was collected from NewsAPI and web scraping. Back translation is used in the data preparation stage to deal with unbalanced data sets and create text variants Grid search is used to find the best hyperparameters for Gradient Boosting and XGBoost. The results of the research are in the form of a machine-learning model. Across all models examined, Gradient Boosting trained on 753 records performed best with an accuracy rate of 82.31% on test data and 84.93% on validation data. This model is able to be used to classify media and then implemented in a web-based interface.**

Keywords*— *adverse media, classification, gradient boosting, website, XGBoost.

I. Introduction

The proliferation of Financial Technology (Fintech) in Indonesia has been notably swift, coinciding with the society's growing acceptance of digital payments. In 2021, there was a 32.5% augmentation in the number of financial technology enterprises relative to the preceding year [1]. The data from 2022 indicates a rise in electronic money transactions, registering a growth of 42.06% in the first quarter alone [2]. Furthermore, the financial technology sector in Indonesia manifests a Compounded Annual Growth Rate (CAGR) of 39% [3].

Technology has given rise to financial technology (fintech) innovations that make it possible to carry out transaction activities quickly and efficiently by utilizing technology [4]. The rapid development of financial technology also increases the risk of money laundering and terrorism financing, especially in this industry. The fast and dynamic nature of financial technology makes financial technology a medium for money laundering and terrorism financing [5]. This causes financial technology companies to be required to identify, verify and analyze user risks in implementing the Anti-Money Laundering and Counter-Terrorism Financing (AML-CTF) program[6].

A start-up company can try to develop similar technology oriented towards solving fraud problems and customer trust problems [7]. In improving customer due diligence services, a machine learning model is needed to classify data from media, especially from online media in Indonesia. The adverse media model created will be used to sort data before it is processed in the customer due diligence system. The data used in making this model comes from scraping on news sites and using NewsAPI as the data source for the model to be made.

The research utilizes XGBoost, known for its ability to handle large data quickly and manage missing values and data imbalance efficiently. However, XGBoost may require more time to train the model and a deeper understanding of its parameters to achieve optimal results. The study will also use Gradient Boosting for adverse media classification to determine if XGBoost outperforms its predecessor.

The ultimate goal is to create a model for classifying news into adverse or non-adverse media categories, assisting a company in processing customer due diligence data sources with higher accuracy. The study aims to address the limitations of previous research, particularly in adverse media classification in Indonesia.

II. Related Works

Several previous studies have focused on text classification related to news data using the AG News dataset. The first algorithm for news text classification is XLNet, achieving an error rate of 4.45% [8]. This research, conducted by Carnegie Mellon University and the Google AI Brain Team, reported an accuracy rate of 85.4%. Another study in text classification used BERT-ITPT-FiT, with an error rate of 4.8% [9]. A subsequent study employed LSTM with a Mixed Objective Function, resulting in an error rate of 4.95% [10]. Another study utilized ULMFiT, achieving an error rate of 5.01% [11].

There are prior studies on text classification using machine learning. One such study employed the Random Forest algorithm with Gradient Boosting for automated complaint categorization [12]. This research successfully applied Random Forest and Gradient Boosting for multi-class

turnitin Page 9 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772
979-8-3503-9174-9/24/$31.00 ©2024 IEEE

text classification, with an accuracy of 73%. The limitation of this study was that most data were classified into a single category due to the words falling into multiple categories.

Another prior study used Decision Tree and XGBoost for COVID-19 vaccine sentiment classification [13]. This research produced an XGBoost model with an accuracy rate of 66%, and a Decision Tree model with an accuracy of 65%. The XGBoost algorithm outperformed the Decision Tree algorithm on an imbalanced dataset. This study had the limitation of an imbalanced dataset and achieved a low accuracy and f1-score of 66%.

The next study was on fake news classification using the XGBoost algorithm [14]. This research achieved an accuracy rate of 92% with an 80:20 dataset ratio. This level of accuracy was obtained based on parameter tuning. The limitation of this study was that the dataset contained only 100 randomly distributed data points. The research may only have the best accuracy on specific datasets and may not produce the same accuracy level on different datasets.

Another study predicted chronic kidney failure by comparing Grid Search and Random Search methods for hyperparameter tuning in the XGBoost algorithm[15]. This study achieved an accuracy rate of 99.29% and an f-measure of 0.99. XGBoost with hyperparameter tuning, z-score normalization, and random oversampling played a significant role in achieving this accuracy. The drawback of this study was the lengthy time taken to find the correct hyperparameters using Grid Search. In addition, the study was not tested on different types of datasets, so the generalization ability of the model was unknown.

Another study predicted and diagnosed future diabetes risk using machine learning. This study produced a model with an accuracy rate of 85% using Gradient Boosting, 77% with Naïve Bayes, and 79% with Logistic Regression[16]. This research achieved high predictive accuracy and used algorithms that are excellent for regression problems, such as Gradient Boosting. This research had several limitations, such as the fact that the data used did not cover the entire population. The research only used BMI and plasma glucose, which are crucial in predicting diabetes. The study has potential overfitting issues because it does not address potential overfitting that can occur in the model.

Based on the results of previous research using machine learning algorithms like Random Forest, convolutional neural network, complement, multinomial naïve bayes, Linear Regression, and XGBoost, there is currently no research on creating models for adverse media, particularly in the Indonesian language. Most of the research on customer due diligence and adverse media is conducted privately. Moreover, this research uses datasets that are not publicly available and requires scraping techniques and API calls for data collection. Therefore, this study will focus on creating a machine learning model used for classifying news, especially online media, whether the content of the news falls into adverse media or not. This study also tests the use of TF-IDF as in research [18], as well as the use of CountVectorizer and Tokenizer, which are commonly used in text data feature extraction. The use of Grid Search in this study aims to determine the best parameters more efficiently. Grid Search facilitates the discovery of optimal hyperparameters in the algorithms used. The parameters will influence the level of accuracy as revealed in studies [14], [15].

III. Methodology

This research will create a machine learning model to classify news and identify whether the news is classified as adverse media or not. The algorithms used in model training are XGBoost and Gradient Boosting. The research methodology is depicted in the flowchart presented in Fig. 1.

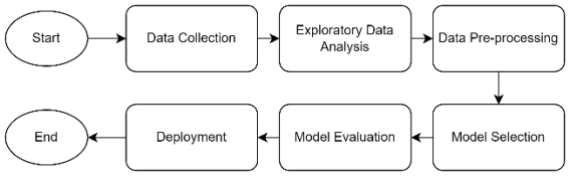


Fig 1. Research Flow

A. Data Collection

Data is gathered using two methods: web scraping and the NewsAPI. The collected news data will be sorted based on keywords that indicate adverse media. The keywords used are based on guidelines from the Indonesian Center for Reporting and Analysis of Financial Transactions (Pusat Pelaporan dan Analisis Transaksi Keuangan (PPATK)). The total data generated is 1,281 records originating from the keywords "korupsi" (corruption), "suap" (bribe) and "narkotika" (narcotics). The data collection process was from 24 January 2022 to 6 February 2022.

B. Exploratory Data Analysis

Exploratory data analysis in this study is by using the univariate analysis method. The analysis is carried out using news title data by calculating the frequency of words from the available dataset. This analysis also looks at the words that appear the most in the news headlines and the words that appear the least often in the news headlines. Apart from frequency, another analysis is by making histograms and word clouds to find out more about the data used.

C. Data Pre-processing

This stage is the stage in the process of cleaning the data and turning the news text into a token matrix. In the process of cleaning the data, the data obtained must be cleaned of some characters that are not needed. After that, a case folding process is carried out which has the goal of turning all letters into lowercase letters [17]. Besides that, the process of cleaning the data also needs to remove stop words so that they don't interfere with the training process in model making. Next stage is to perform the feature extraction process using several types, namely TF-IDF, Tokenizer, and CountVectorizer.

D. Model Selection

According to the previous research, it stated that there is no model that is generally suitable for certain data and purposes. The process of finding a model that fits the dataset that is owned is done by finding the most suitable model or in accordance with the characteristics of the existing data [19]. Model selection is used to get the best model according to the data provided. The model selection process in XGBoost and Gradient Boosting is done by looking for the best hyperparameter with the best performance. The process of finding the best hyperparameters in XGBoost and Gradient Boosting can be done using a grid search. XGBoost has several hyperparameters that can be used. Some of these

parameters include learning_rate, max_depth, min_child_weight and gamma [18].

E. Model Evaluation

Model evaluation is done by comparing the average precision and recall values between the train data and test data or validation data. If the difference exceeds 15%, the model is overfitting. If the difference is less than 1%, the model is underfitting. The optimal model has a difference between 1% and 5%. The calculation process from the difference between precision and recall and generates Good Fit, Overfit and Underfit labels. The resulting model is as shown in table 1.

TABLE I. MODEL RESULTS

Dataset	Algorithm	Accuracy	Fit 5%	Fit 15%
753	Gradient Boosting - CountVectorizer	82.31%	Overfit	Good Fit
1.281	XGBoost – TF-IDF	78.54%	Overfit	Good Fit

The top-performing model derived from XGBoost demonstrates a robust performance, incorporates TF-IDF text processing, and makes use of a dataset with 1,281 instances. With regard to test data and validation data, this model achieves an admirable accuracy of 78.54% and 82.52%, respectively. It was chosen because it strikes a better balance between model stability and performance. In particular, it performs better than alternatives in terms of metrics like precision, recall, F1-score, and overall accuracy. Due to overfitting concerns in models using CountVectorizer under various dataset conditions, a problem that was particularly noticeable within the XGBoost framework, the use of TF-IDF emerged as a viable option.

Moving on to the area of Gradient Boosting, the top model uses CountVectorizer for text processing and runs on a dataset with 753 instances. This model stands out because it performs at its best across a variety of evaluative criteria, such as accuracy, precision, recall, and F1-score. Despite exhibiting slight overfitting symptoms—encountered at the 5% limit but avoided at the 15%—the model maintains its usefulness in practice due to its reliable and outstanding performance. It produces test data accuracy rates of 82.31% and validation data accuracy rates of 84.93%.

The Gradient Boosting model, distinguished by its accuracy rate of 82.31% on test data and 84.93% on validation data, is chosen for the deployment phase after a thorough comparison of the two models. The deployed model will be used to categorize news titles into negative media categories, demonstrating a robust use case for the model. The selected model is ready for integration with the Flask framework at the deployment stage.

F. Deployment

Deployment is carried out using a website platform connected to the Adverse Media Classification API. The backend of Adverse Media Classification is built using the Python Flask framework. The Flask framework handles request from the frontend, routing them to the designed endpoints for processing. The backend subsequently validates data and executes the specified algorithm. The backend invokes the model and the responses from the model then converted into JSON values to utilization on the frontend page. The backend requires several parameters in using the Adverse Media Classification model. The parameters used are title and algorithm. The title parameter will be used as the title of the news that will be analyzed with the model that has been made. Algorithm parameters are used to select the algorithm used in the classification process. The website platform can be accessed via the frontend once it has been deployed. Users can access the website from anywhere, provided they have an internet connection and the platform is deployed on the server.

IV. RESULTS AND DISCUSSION

This study offers a comparative evaluation of machine learning models for media classification with a specific focus on adverse and non-adverse categories. Gradient Boosting and XGBoost models were investigated thoroughly, with the former exhibiting superiority in terms of accuracy, precision, recall, f1-score, and train-test differences. The achieved accuracy rate for most models exceeded 70%.

The study employed diverse text processing techniques such as TF-IDF, CountVectorizer, and Tokenizer, yielding varying results. Despite the Tokenizer's high accuracy in certain models, its performance was subpar on test data due to overfitting, suggesting the necessity for further investigation. Interestingly, the study discovered that there is no direct correlation between data volume and model accuracy. This suggests that increasing the amount of data does not necessarily result in decreased accuracy. The research also underscored the idea that the accuracy achieved in machine learning models may not always represent real-world accuracy due to limitations such as data availability and variations in word usage.

These findings align with previous studies. For example, the Gradient Boosting model outperformed the Random Forest in a study titled "Automatic Complaints Categorization Using Random Forest and Gradient Boosting"[12], a result that was replicated in the current study. Likewise, the effectiveness of Grid Search, which was used in the current study, was confirmed in "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure"[15]. The practical application of this research is the creation of a website for news title analysis. The study recommends continuous monitoring of model performance to guarantee its applicability. Furthermore, there is potential for future enhancements and the integration of the model into company's customer due diligence systems via an API.

V. CONCLUSION

The accuracy of the produced model in classifying adverse media and non-adverse media becomes a measure of the effectiveness of the generated model. The classification model was created using the Python programming language, applying the XGBoost and Gradient Boosting algorithms. The result of the created model will be embedded in a Flask framework, and users can utilize this model through a web page or API call. The model used is a Gradient Boosting model with an accuracy rate of 82.31% on test data and 84.93% on validation data.

The model from this research can be used to classify adverse media on new, which this classification process will simplify the task of grouping the news that will be used in the customer due diligence system. This model is already usable

in the classification process but still needs further development due to the limitations of the data used.

The generated model is a model from the Gradient Boosting algorithm with an accuracy value of 82.31% on test data and 84.93% from validation data. Although these models have a sufficient accuracy rate, it should be remembered that the limitations of the data used in this study affect the model's performance. The amount of data used is only as much as 753, which is still considered insufficient. Therefore, to improve the performance of this model, more data and more complex word combination variations are required.

In addition, these models may not effectively deal with new data that has never been taught during training. The variation of circulating news titles is very diverse, and this model may not be able to handle all these variations with a high accuracy rate. Therefore, this research recognizes these limitations and suggests using more data in future research to improve model performance.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Universitas Multimedia Nusantara for their invaluable support, which played a pivotal role in the successful completion of this research endeavor. Their substantial contribution was instrumental in achieving our objectives, and we are deeply grateful for their unwavering assistance.

REFERENCES

[1] A. Karnadi, "Jumlah Fintech di Indonesia Terus Meningkat hingga 2021." Accessed: Mar. 15, 2023. [Online]. Available: https://dataindonesia.id/digital/detail/jumlah-fintech-di-indonesia-terus-meningkat-hingga-2021

[2] D. F. Rahman, "Transaksi Keuangan Digital Tumbuh Pesat pada Triwulan I 2022." Accessed: Mar. 15, 2023. [Online]. Available: https://databoks.katadata.co.id/datapublish/2022/04/19/transaksi-keuangan-digital-tumbuh-pesat-pada-triwulan-i-2022

[3] "Bulan Fintech Nasional, 1,5 Juta Masyarakat Berpartisipasi dan Dapatkan Edukasi Fintech dari Pemerintah, Asosiasi, dan Pelaku Industri." Accessed: Mar. 15, 2023. [Online]. Available: https://www.bi.go.id/id/publikasi/ruang-media/news-release/Pages/sp_2433922.aspx

[4] E. N. Sugiarti, N. Diana, and M. C. Mawardi, "PERAN FINTECH DALAM MENINGKATKAN LITERASI KEUANGAN PADA USAHA MIKRO KECIL MENENGAH DI MALANG," e_Jurnal Ilmiah Riset Akuntansi, vol. 8, no. 04, Jul. 2019, Accessed: Jan. 07, 2023. [Online]. Available: http://riset.unisma.ac.id/index.php/jra/article/view/4038

[5] H. M. A. Nasution, "HATI-HATI PENCUCIAN UANG DI INDUSTRI FINTECH! - PUSAT PELAPORAN DAN ANALISIS TRANSAKSI KEUANGAN." Accessed: Mar. 15, 2023. [Online]. Available: https://www.ppatk.go.id/siaran_pers/read/969/hati-hati-pencucian-uang-diindustri-fintech.html

[6] F. Annisa and P. R. Putri, "PENERAPAN PROGRAM APU PPT UNTUK MENCEGAH PENCUCIAN UANG DAN PENDANAAN TERORISME PADA INDUSTRI FINTECH," ADIL: Jurnal Hukum, vol. 11, no. 2, Dec. 2020, doi: 10.33476/AJL.

[7] "Tentang Kami — Kredibel." Accessed: Jan. 07, 2023. [Online]. Available: https://www.kredibel.co.id/about

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.08237

[9] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," May 2019, [Online]. Available: http://arxiv.org/abs/1905.05583

[10] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function," Sep. 2020, doi: 10.1609/aaai.v33i01.33016940.

[11] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.06146

[12] M. T. Anwar, "Automatic Complaints Categorization Using Random Forest and Gradient Boosting," Advance Sustainable Science, Engineering and Technology, vol. 3, no. 1, p. 0210106, Apr. 2021, doi: 10.26877/asset.v3i1.8460.

[13] H. H. Sinaga and S. Agustian, "Pebandingan Metode Decision Tree dan XGBoost untuk Klasifikasi Sentimen Vaksin Covid-19 di Twitter," Jurnal Nasional Teknologi dan Sistem Informasi, vol. 8, no. 3, pp. 107–114, Dec. 2022, doi: 10.25077/TEKNOSI.v8i3.2022.107-114.

[14] J. P. Haumahu, S. D. H. Permana, and Y. Yaddarabullah, "Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost)," IOP Conf Ser Mater Sci Eng, vol. 1098, no. 5, p. 052081, Mar. 2021, doi: 10.1088/1757-899x/1098/5/052081.

[15] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," International Journal of Intelligent Engineering and Systems, vol. 14, no. 6, pp. 198–207, Dec. 2021, doi: 10.22266/ijies2021.1231.19.

[16] R. Birjais, A. K. Mourya, R. Chauhan, and H. Kaur, "Prediction and diagnosis of future diabetes risk: a machine learning approach," SN Appl Sci, vol. 1, no. 9, Sep. 2019, doi: 10.1007/s42452-019-1117-9.

[17] M. N. Randhika, J. C. Young, A. Suryadibrata, and H. Mandala, "Implementasi Algoritma Complement dan Multinomial Naïve Bayes Classifier Pada Klasifikasi Kategori Berita Media Online," Ultimatics : Jurnal Teknik Informatika, vol. 13, no. 1, pp. 19–25, Jun. 2021, doi: 10.31937/TI.V13I1.1921.

[18] Y. Wang and X. S. Ni, "A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization," International Journal of Database Management Systems, vol. 11, no. 01, pp. 01–17, Jan. 2019, doi: 10.48550/arxiv.1901.08433.

[19] Frentzen, J. Wiratama and R. S. Oetama, "KNN And Naïve Bayes Algorithms for Improving Prediction of Indonesian Film Ratings using Feature Selection Techniques," 2023 4th International Conference on Big Data Analytics and Practices (IBDAP), Bangkok, Thailand, 2023, pp. 1-6, doi: 10.1109/IBDAP58581.2023.10271977.

turnitin Page 13 of 204 - Integrity Submission Submission ID

CERTIFICATE OF PARTICIPATION

This certificate is proudly presented to

Reza Juliandri; Monika Evelin Johan; Jansen Wiratama; Samuel Ady Sanjaya

For presenting a paper entitled

Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms

The 5th International Conference on Big Data Analytics and Practices (IBDAP 2024)
held on August 23 - 25, 2024
Organized by Big Data Institute (Public Organization)

Assoc. Prof. Tiranee Achalakul, PhD
President of Big Data Institute

IBDAP

turnitin Page 13 of 204 - Integrity Submission Submission ID

ID: 1571039738



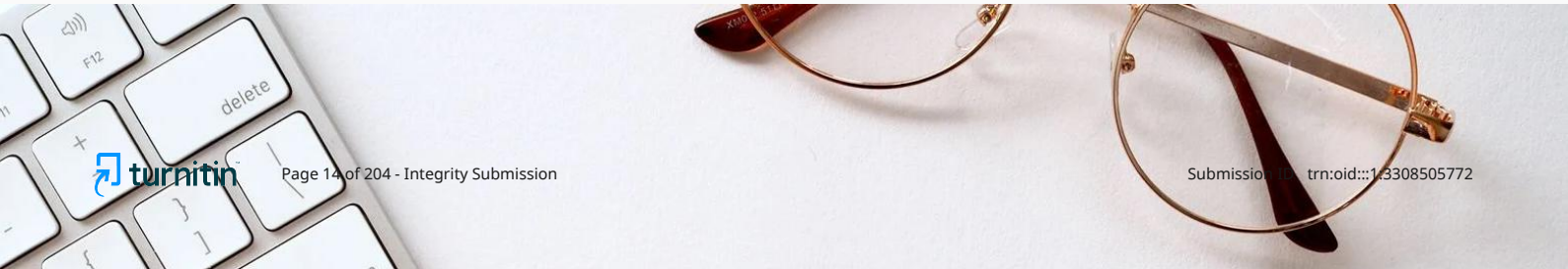
IBDAP 2024

THE 5TH INTERNATIONAL CONFERENCE ON BIG DATA ANALYTICS AND PRACTICES (IBDAP2024)

BANGKOK, THAILAND
AUGUST 23 – 25, 2024

ORGANIZED BY
BIG DATA INSTITUTE (PUBLIC ORGANIZATION)

3 Big Data Analytics and Mining
- Algorithms and systems for big data search and analytics
- Machine learning for big data
- Predictive analytics and simulation
- Big data visualization and interactive data exploration
- Big data mining applications
- Knowledge extraction, discovery, analysis, and presentation
- Big Data Platforms and Technologies
- Big data processing frameworks and technologies
- Big data services and application development methods and tools
- Big data quality evaluation and assurance technologies
- Big data system reliability, dependability, and availability
- Open-source development and technology for big data
- Big Data as a Service (BDaaS) platform and technologies
- Big Data and Machine Learning Applications and Experiences
- Innovative big data applications and services
- Big data analytics in the public sector
- Large-scale recommendation systems
- Link and graph mining, social network mining
- Mobility and big data
- Stream data mining
- Real-world and large-scale practices of big data





Proceedings of

The 5th International Conference on Big Data Analytics and Practices (IBDAP 2024)

August 23-25, 2024

Bangkok, Thailand

Big Data Institute (Public Organization)

Ministry of Digital Economy and Society

ISBN (e-Book): 978-616-8333-05-1



Organized by

BIG DATA INSTITUTE
สถาบันข้อมูลขนาดใหญ่

Sponsored

กระทรวงดิจิทัล
เพื่อเศรษฐกิจและสังคม

depa

Supported by

IEEE

IEEE
THAILAND SECTION

IEEE
Signal
Processing
Society
THAILAND CHAPTER

สคทส.
CITT
สมาคมสภาคณบดีคณะเทคโนโลยีสารสนเทศ

©IBDAP 2024 - BDI

The 5th International Conference on Big Data Analytics and Practices (IBDAP 2024)

Big Data analytics and smart computing are emerging fields that have recently drawn much attention from research communities in computer science, information technology, social science, and many other disciplines.

The International Conference on Big Data Analytics and Practices (IBDAP 2023), initiated by Big Data Institute (Public Organization), provides an international forum for exchanging ideas and information on current progresses, challenges, research results, system developments, and practical experiences in these emerging fields.

IBDAP calls for submission of work from a wide range of domains, including but not limited to smart systems (e.g. in cities, automobiles, farms, etc.), cyber-physical systems, Internet of Things applications, healthcare, social networks, and media. Authors are encouraged to submit papers covering big data topics, such as service-oriented technologies, machine learning, predictive analytics, data modeling, system architectures, data mining, and simulation.

Published by

Big Data Institute (Public Organization)
Ministry of Digital Economy and Society
ISBN (e-Book): 978-616-8333-05-1

Website IBDAP: https://www.ibdap.org/

Table of Contents

Conference Organizers 01

Keynote Speaker 07

Program Schedule 08

Regular Papers

Web URLs Phishing Detection Model with Random Forest Algorithm
Aulia Kharisma Putri; Jansen Wiratama; Samuel Ady Sanjaya; Santo Fernandi Wijaya; Monika Evelin Johan; Ahmad Faza 11

Convolutional Vision Transformer Modeling for Spectrogram Image Processing in the Detection of North Atlantic Right Whales Up-Call
Siti Ummi Masruroh; Muhammad Destamal Junas; Khodijah Hulliyah; Husni Teja Sukmana; Rizka Amalia Putri; Saepul Aripiyanto 16

Data Veracity Analysis in Social Medias - A Review
Bellouqi Mohamed Amine; Ismail Assayad 22

The Prospective Threat Vector of a Bounded and Controllable Optimized Computational Approach for Spatio-Temporal Knowledge Graph Completion
Steve Chan 28

Addressing Gender Bias: A Fundamental Approach to AI in Mental Health
Karikarn Chansiri; Xinyu Wei; Ka Ho Brian Chor 35

An Analysis of Synthetic Data for Improving Performance of Skeleton-Based Fall down Detection Models
Park Jimin; Bongjun Kim; Junho Jeong 41

Development of Open-Source Big Data Technology Using Project Management to Addressing the Complexity in ERP Implementation
Santo Fernandi Wijaya; Angelina Ervina Jeanette Egeten; Jansen Wiratama 45

An Improvement on Exploration Step of Whale Optimization Algorithm with Levy Distribution for Classification Problems
Sakkayaphop Pravesjit; Krittika Kantawong; Natdanai Kamkhad; Saksit Sabaiporn; Jantawan Monchanuan; Duangjai Jitkongchuen; Arit Thammano; Panchit Longpradit 51

Table of Contents

Regular Papers

Modification of Sand Cat Swarm Optimization for Classification Problems
Sakkayaphop Pravesjit; Krittika Kantawong; Sathien Hunta; Duangjai Jitkongchuen; Arit Thammano; Panchit Longpradit 57

Basking Behavior in Cold-Blooded and Warm-Blooded Reptiles: A Systematic Review of Interspecies Treatment
Chanatkit Harnnuengnit; Sarochar Khambuo; Puthyrom Tep 62

Elevating Air Quality Forecasting: Integrating Hybrid Clustering Techniques with Long Short-Term Memory Networks
Irfan Fari Ramadhan; Samuel Ady Sanjaya 68

Transfer Learning Approach for Rainfall Class Amount Prediction Using Uganda's Lake Victoria Basin Weather Dataset
Andrew Gahwera; Odongo Steven Eyobu; Isaac Mugume 76

Forecasting the NBA's Most Valuable Player: A Regression Analysis Approach
Arnando Harlianto; Johan Setiawan 82

Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms
Reza Juliandri; Monika Evelin Johan; Jansen Wiratama; Samuel Ady Sanjaya 89

Evaluating GRNN, Decision Tree, and Random Forest: A Gas Turbine Emission Prediction Comparative Study
Rudy Winarto; Mauridhi Hery Purnomo; Wiwik Anggraeni 93

Hybrid PSO-CNN Model for Cross-Domain Adaptation Sentiment Analysis
Ummu Fatihah Binti Mohd Bahrin; Hamidah Jantan 101

Enhanced Pooling Technique in Convolutional Neural Networks Model for Classification of Magnoliophyta Plant DNA Barcodes
Lilibeth P. Coronel; Arnel C. Fajardo; Ruji Medina 106

Flood Susceptibility Mapping Using Publicly Available Big Data with Google Earth Engine and Deep Learning Algorithms
Sackdavong Mangkhaseum; Yogesh Bhattarai; Sunil Duwal; Akitoshi Hanazawa 111

Enhancing Short Text Semantic Similarity Measurement Using Pretrained Word Embeddings and Big Data
Supakpong Jinarat; Ratchakoon Pruengkarn 117

Enhancing Durian Cultivation Efficiency Through Data-Driven Smart Farming Using Cluster Analysis and Machine Learning
Pattharaporn Thongnim; Jakkrapan Sreekajon; Thanaphon Phukseng 121

Table of Contents

Regular Papers

Predicting China's Marriage Rate: Dual Machine Learning (DML) for Causal Inference Using XGBoost, LightGBM, CatBoost, GBDT
Deyu Zhang; Worasak Rueangsirarak; Surapong Uttama 127

Comparison of the Statistical and Autoencoder Approach for Anomaly Detection in Big Data
Barasha Mali 133

Grapevine Leaf Disease Classification Using Deep Convolutional Neural Networks
Md Al-Imran; Sajid Faysal Fahim; Sanjida Simla; Fatin Hasnat Shakib; Tokiuddin Ahmed; Sarwar Jahan 137

PCB Surface Defect Detection Using Defect-Centered Image Generation and Optimized YOLOv8 Architecture
Thongpan Supong; Thanapat Kangkachit; Duangjai Jitkongchuen 143

Detection of Infective Juvenile Stage of Entomopathogenic Nematodes Using Deep Learning
Uthai Phuyued; Thanapat Kangkachit; Duangjai Jitkongchuen 149

Effect of Sliding Window Sizes on Sensor-Based Human Activity Recognition Using Smartwatch Sensors and Deep Learning Approaches
Sakorn Mekruksavanich; Ponnipa Jantawong; Anuchit Jitpattanakul 155

Integrating In-Ear Wearable Sensors with Deep Learning for Head and Facial Movement Analysis
Sakorn Mekruksavanich; Ponnipa Jantawong; Anuchit Jitpattanakul 161

Ensemble Deep Learning Network for Enhancing Performances of Sensor-Based Physical Activity Recognition Based on IMU Sensor Data
Sakorn Mekruksavanich; Ponnipa Jantawong; Anuchit Jitpattanakul 167

Design and Development of a Vertical Garden Station with Plants and an Automatic Fogging System for PM2.5 Reduction
Kanteera Mekruksavanich; Natthayada Thamchaikul; Parachaya Muentabutra; Nongnapas Sutthipornmaneewat 173

Reel Tower Control Using Machine Learning
MyeongSu Jeong; ChangSoo Moon; JaeHoon Chung 178

Author Index 1

3

International Advisory Committees

Simon See (Shanghai Jiao Tong University, Singapore)

Lerwen Liu (National University of Singapore, Singapore)

Kaoru Sezaki (University of Tokyo, Japan)

Pierre Vande Vyvre (CERN, Switzerland)

General Chair

Tiranee Achalakul (BDI, Thailand)

General Co-Chair

Krittika Kantawong (University of Phayao, Thailand, IEEE SPS Chapter) #94103612

Organizing Community and Committees

IEEE Signal Processing Society Thailand Chapter

The Association of Council of IT Deans

Technical Program Chair

Duangjai Jitkonghcheun (BDI, Thailand) #98211670

Datchakorn Tancharoen (PIM, Thailand) #41635099

Parisut Jitpakdee (BDI, Thailand)

Technical Program Co-Chair

Punnarumol Temdee (Mae Fah Luang University, Thailand, IEEE SPS Chapter) #94471090

Pradorn Sureephong (Chiangmai University, Thailand, IEEE SPS Chapter) #98063511

Publication Chair

Paween Khoenkaw (Maejo University, Thailand, IEEE SPS Chapter) #94052729

Santitham Prom-on (KMUTT, Thailand) #80458455

Program Committees

Khwansiri Sirimangkhala (BDI, Thailand)
Angkhana Prommarat (BDI, Thailand)
Pirommas Techitnutsarut (BDI, Thailand)
Prof. Dusit Niyato (Nanyang Technological University, Singapore)
Jiawen Kang (Guangdong University of Technology, China)
Nguyen Van Huynh (University of Liverpool, United Kingdom)
Xiong Zehui (Singapore University of Technology and Design, Singapore)
Ujjwal Kumar Deb (Chittagong University of Engineering and Technology, Bangladesh)
Nguyen Dang Hoa Nghiem (Can Tho University of Technology, Vietnam)
Paven Thuat Do (Vietnam Blockchain Association, Vietnam)
Lin Mongkolsery (Institute of Technology of Cambodia, Cambodia)
Nghi Truong (Sasin Graduate Institute of Business Administration of Chulalongkorn University, Thailand)
Stefano Starita (Sasin Graduate Institute of Business Administration of Chulalongkorn University, Thailand)
Anne Cheng (Supercharge Lab, United States)
Stephanie Sy (Thinking Machines Data Science, Singapore)
Sally Goldin (CMKL University, Thailand)
Masanori Kondo (Asia-Pacific Telecommunity, Thailand)

Organizing Committees

Natawut Nupairoj (Chula, Thailand)
Sukumal Kitisin (KU, Thailand)
Khajonpong Akkarajitsakul (KMUTT, Thailand)
Nawaporn Wisitpongphan (KMUTNB, Thailand)
Akkradach Watcharapupong (KMITL, Thailand)
Pakorn Ubolkosold (BU, Thailand)
Jirapon Sunkpho (TU, Thailand)
Pagaporn Pengsart (MU, Thailand)
Chetneti Srisaan (RSU, Thailand)
Surachai Thongkaew (SPU, Thailand)
Chatchai Powthongchin (SU, Thailand)
Pichaya Tandayya (PSU, Thailand)
Soonthorn Pibulcharoensit (ABAC, Thailand)
Nattagit Jiteurtragool, (TNI, Thailand)
Arnond Sakworawich (NIDA, Thailand)
Chaiyaporn Khemapatapan (DPU, Thailand)

Technical Program Committees [1]

Adisak Intana (Prince of Songkla University), Thailand
Ajchara Phu-ang (Thammasat University), Thailand
Akadej Udomchaiporn (King Mongkut's Institute of Technology Ladkrabang), Thailand
Akkradach Watcharapupong (King Mongkut's Institute of Technology Ladkrabang), Thailand
Amarita Ritthipakdee (Phranakorn Rajabhat University), Thailand
Anantaporn Hanskunatai (King Mongkut's Institute of Technology Ladkrabang), Thailand
Angkhana Prommarat (Big Data Institute (Public Organization)), Thailand
Arnond Sakworawich (National Institute of Development Administration), Thailand
Artit Sagoolmuang (Big Data Institute (Public Organization)), Thailand
Assadarat Khurat (Mahidol University), Thailand
Aziz Nanthaamornphong (Prince of Songkla University), Thailand
Boonsit Yimwadsana (Mahidol University), Thailand
Chadaporn Keatmanee (Thai-Nichi Institute of Technology), Thailand
Chakree Teekapakvisit (King Mongkut's Institute of Technology Ladkrabang), Thailand
Chaloemphon Sirikayon (Dhurakij Pundit University), Thailand
Chatchai Powthongchin (Silpakorn University), Thailand
Chetneti Srisaan (Rangsit University), Thailand
Chidchanok Choksuchat (Prince of Songkla University), Thailand
Cholrit Luangjinda (Thai-Nichi Institute of Technology), Thailand
Chuleekorn Nuansomsri (Rangsit University), Thailand
Datchakorn Tancharoen (Panyapiwat Institute of Management (PIM)), Thailand
Dechanuchit Katanyutaveetip (Siam University), Thailand
Eakasit Pacharawongsakda (Dhurakij Pundit University), Thailand
Ekarat Rattagan (NIDA), Thailand
Hutchatai Chanlekha (Kasetsart University), Thailand
Jian Qu (Panyapiwat Institute of Management (PIM)), Thailand
Jirapon Sunkpho (Thammasat University), Thailand
Kanjana Laosen (Prince of Songkla University), Thailand
Kanticha Kittipeerachon (Thai-Nichi Institute of Technology), Thailand
Karin Sumongkayothin (Mahidol University), Japan
Karn Yongsiriwit (Rangsit University), Thailand
Kasem Thiptarajan (Thai-Nichi Institute of Technology), Thailand
Kasikrit Damkliang (Prince of Songkhla University), Thailand
Kawiwat Amnatchotiphan (Thai-Nichi Institute of Technology), Thailand
Khajonpong Akkarajitsakul (King Mongkut's University of Technology Thonburi), Thailand
Khwansiri Sirimangkhala (Big Data Institute (Public Organization)), Thailand
Kitsiri Chochiang (Prince of Songkla University), Thailand
Kitsuchart Pasupa (King Mongkut's Institute of Technology Ladkrabang), Thailand
Kriangkrai Limthong (Bangkok University), Thailand
Krittika Kantawong (University of Phayao), Thailand

Technical Program Committees [2]

Kulwadee Somboonviwat (Kasetsart University Sriracha), Thailand
Kwankamon Dittakan (Prince of Songkla University), Thailand
5 Maleerat Sodanil (King Mongkut's University of Technology North Bangkok), Thailand
Nantika Prinyapol (Dhurakij Pundit University), Thailand
Narit Hnoohom (Mahidol University), Thailand
Narongdech Keeratipranon (Dhurakij Pundit University), Thailand
Narungsun Wilaisakoolyong (Thai-Nichi Institute of Technology), Thailand
Natawut Nupairoj (Chulalongkorn University), Thailand
5 Nattagit Jiteurtragool (King Mongkut's University of Technology North Bangkok), Thailand
Nattagit Jiteurtragool (Thai-Nichi Institute of Technology), Thailand
Nattapol Aunsri (Mae Fah Luang University), Thailand
Nattapong Sanchan (Bangkok University), Thailand
5 Nawaporn Wisitpongphan (King Mongkut's University of Technology North Bangkok), Thailand
Nilubon Kurubanjerdjit (Mae Fah Luang University), Thailand
Nittaya Kerdyam (Siam University), Thailand
Nopporn Chotikakamthorn (King Mongkut`s Institute of Technology Ladkrabang, Bangkok), Thailand
Pagaporn Pengsart (Mahidol University), Thailand
Pakachart Puttipakorn (Thai-Nichi Institute of Technology), Thailand
Pakapan Limtrairut (Bangkok University), Thailand
Pakorn Ubolkosold (Bangkok University), Thailand
Panita Thusaranon (Dhurakij Pundit University), Thailand
Paralee Maneerat (Chulalongkorn University), Thailand
36 Peeradon Samasiri (King Mongkut's University of Technology Thonburi), Thailand
Phaisarn Sudwilai (Advanced Magnetic and Motor Drive Research Laboratory (AMDRL)), Thailand
Phayung Meesad (King Montkut's University of Technology North Bangkok), Thailand
Pichaya Tandayya (Prince of Songkla University), Thailand
Pichit Sukchareonpong (Thai-Nichi Institute of Technology), Thailand
Pirommas Techitnutsarut (Big Data Institute (Public Organization)), Thailand
Pornavalai Chotipat (King Mongkut's Insititute of Technology Ladkrabang), Thailand
Pornthep Rojanavasu (University of Phayao), Thailand
Prajak Chertchom (Thai-Nichi Institute of Technology), Thailand
Prajaks Jitngernmadan (Burapha University), Thailand
Pramuk Boonsieng (Thai-Nichi Institute of Technology), Thailand
Pranisa Israsena (Thai Nichi Institute of Technology), Thailand
Preecha Tangworakitthaworn (Mahidol University), Thailand
Ratchakoon Pruengkarn (Dhurakij Pundit University), Thailand
Rattana Wetprasit (Prince of Songkla University), Thailand
Saiyan Saiyod (Khon Kaen University), Thailand
Sakorn Mekruksavanich (University of Phayao), Thailand
Salil Boonbrahm (Walailak University), Thailand

Technical Program Committees [3]

Sangsuree Vasupongayya (Prince of Songkla University), Thailand
Saprangsit Mruetusatorn (Thai-Nichi Institute of Technology), Thailand
Saranthorn Phusingha (Big Data Institute (Public Organization)), Thailand
Sarawut Ramjan (Thammasat University), Thailand
Sarayut Nonsiri (Thai-Nichi Institute of Technology), Thailand
Saromporn Charoenpit (Thai-Nichi Institute of Technology), Thailand
Silada Intarasothonchun (Khon Kaen University), Thailand
Sinchai Kamolphiwong (Prince of Songkla University), Thailand
Sirion Vittayakorn (King Mongkut's Institute of Technology Ladkrabang), Thailand
Songsri Tangsripairoj (Mahidol University), Thailand
Soontarin Nupap (Mae Fah Luang University), Thailand
Soonthorn Pibulcharoensit (Assumption University), Thailand
Suchart Khummanee (Mahasarakham University), Thailand
Sukumal Kitisin (Kasetsart University), Thailand
Supaporn Chairungsee (Walailak University), Thailand
Suppakarn Chansareewittaya (Mae Fah Luang University), Thailand
Suppat Rungraungsilp (Walailak University), Thailand
Surachai Thongkaew (Sripatum University), Thailand
Surangkana Rawungyot (University of Phayao), Thailand
Surapong Uttama (Mae Fah Luang University), Thailand
Suttisak Jantavongso (Rangsit University), Thailand
Tanapon Jensuttiwetchakul (King Montkut's University of Technology North Bangkok), Thailand
Taravichet Titijaroonroj (King Mongkut's Institute of Technology Ladkrabang), Thailand
Thana Udomsripaiboon (University of Phayao), Thailand
Thanapat Kangkachit (Dhurakij Pundit University), Thailand
Thanapon Noraset (Mahidol University), Thailand
Thanathip Limna (Prince of Songkla University), Thailand
Theekapun Charoenpong (Srinakharinwirot University), Thailand
Thitinan Tantidham (Mahidol University), Thailand
Thitiporn Lertrusdachakul (Thai-Nichi Institute of Technology), Thailand
Thongchai Kaewkiriya (Panyapiwat Institute of Management), Thailand
Toshiaki Kondo (Sirindhorn International Institute of Technology), Thailand
Ungsumalee Suttapakti (Burapha University), Thailand
Vanvisa Chutchavong (King Mongkut's Institute of Technology Ladkrabang), Thailand
Vasin Chooprayoon (Rangsit University), Thailand
Virach Sornlertlamvanich (Musashino University), Japan
Vithida Chongsuphajaisiddhi (King Mongkut's University of Technology Thonburi), Thailand
Vorapitchaya Rabiablok (Big Data Institute (Public Organization)), Thailand
Warakorn Srichavengsup (Thai-Nichi Institute of Technology), Thailand
Warangkhana Kimpan (King Mongkut's Institute of Technology Ladkrabang), Thailand

Technical Program Committees [4]

Waraporn Jirapanthong (Dhurakij Pundit University), Thailand
Wasimon Panichpattanakul (Prince of Songkla University), Thailand
Watchareewan Jitsakul (King Mongkut's University of Technology North Bangkok), Thailand
Weerapat Satitkanitkul (Big Data Institute (Public Organization)), Thailand
Weerawut Thanhikam (Panyapiwat Institute of Management), Thailand
Wimol San-Um (Thai-Nichi Institute of Technology), Thailand
Wiphada Wettayaprasit (Prince of Songkla University), Thailand
Wisarn Patchoo (Bangkok University), Thailand
Worapan Kusakunniran (Mahidol University), Thailand
Worapat Paireekreng (Dhurakij Pundit University), Thailand
Worasak Rueangsirarak (Mae Fah Luang University), Thailand
Woratat Makasiranondh (Rangsit University), Thailand
Wudhichart Sawangphol (Mahidol University), Thailand

Keynote Speakers

Opening Ceremony



Assoc. Prof. Tiranee Achalakul, Ph.D.
President of Big Data Institute
(Public Organization)

Title:
Powering Future with Data and AI

Keynote Session 1



Krin Chinprasatsak
CEO & AI researcher at Made by AI

Title:
Visualization of Mathematics behind Deep Learning

Keynote Session 2



Dr.Sanparith Marukatat
Committee and Head of Academic at AIAT / Senior Researcher at NECTEC

Title:
Recent Advances in AI

Keynote Session 3



Mr. Supasate Vorathammathorn
Super AI Engineer Gold Medal

Title:
Prompt Engineer for HR Analytics

The 5th International Conference
on Big Data Analytics and Practices (IBDAP 2024)

BIG DATA INSTITUTE

©IBDAP 2024 - BDI

turnitin Page 26 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Program Schedule

Day 1 Friday August 23, 2024

TIME	EVENT
08.30 - 15.00	Registration
09.00 - 09.45	**Opening Ceremony** Assoc. Prof. Tiranee Achalakul, Ph.D. President of Big Data Institute (Public Organization) **Title: Powering Future with Data and AI**
09.45 - 10.45	**Keynote Session 1** Keynote Speaker: Mr.Krin Chinprasatsak CEO & AI researcher at Made by AI **Title: Visualization of Mathematics behind Deep Learning**
10.45 - 11.00	Session Break
11.00 - 12.00	**Keynote Session 2** Keynote Speaker: Dr.Sanparith Marukatat Committee and Head of Academic at AIAT & Senior Researcher at NECTEC **Title: Recent Advances in AI**
12.00 - 13.30	Lunch
13.30 - 14.45	Oral Session 1
13.30 – 13.45	Web URLs Phishing Detection Model with Random Forest Algorithm
13.45 – 14.00	Development of Open-Source Big Data Technology Using Project Management to Addressing the Complexity in ERP Implementation
14.00 – 14.15	Basking Behavior in Cold-Blooded and Warm-Blooded Reptiles: A Systematic Review of Interspecies Treatment
14.15 – 14.30	Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms
14.30 – 14.45	Comparison of the Statistical and Autoencoder Approach for Anomaly Detection in Big Data
14.45 – 15.00	Session Break
15.00 – 16.15	Oral Session 2
15.00 – 15.15	The Prospective Threat Vector of a Bounded and Controllable Optimized Computational Approach for Spatio-Temporal Knowledge Graph Completion
15.15 – 15.30	Enhanced Pooling Technique in Convolutional Neural Networks Model for Classification of Magnoliophyta Plant DNA Barcodes
15.30 – 15.45	Flood Susceptibility Mapping Using Publicly Available Big Data with Google Earth Engine and Deep Learning Algorithms
15.45 – 16.00	PCB Surface Defect Detection Using Defect-Centered Image Generation and Optimized YOLOv8 Architecture
16.00 – 16.15	Detection of Infective Juvenile Stage of Entomopathogenic Nematodes Using Deep Learning
18.00 – 21.00	Banquet

Program Schedule

Day 2 Saturday August 24, 2024

TIME	EVENT
08.30 - 15.00	Registration
09.00 - 10.30	**Practical Workshop** Speaker: Mr. Supasate Vorathammathorn Super AI Engineer Gold Medal **Title: Prompt Engineer for HR Analytics**
10.30 - 10.45	Session Break
10.45 - 12.00	**Practical Workshop (Cont.)**
12.00 - 13.30	Lunch
13.30 - 14.45	Oral Session 3
13.30 – 13.45	Forecasting the NBA's Most Valuable Player: A Regression Analysis Approach
13.45 – 14.00	Enhancing Short Text Semantic Similarity Measurement Using Pretrained Word Embeddings and Big Data
14.00 – 14.15	Enhancing Durian Cultivation Efficiency Through Data-Driven Smart Farming Using Cluster Analysis and Machine Learning
14.15 – 14.30	Predicting China's Marriage Rate: Dual Machine Learning (DML) for Causal Inference Using XGBoost, LightGBM, CatBoost, GBDT
14.30 – 14.45	Reel Tower Control Using Machine Learning
14.45 – 15.00	Session Break
15.00 – 16.00	Oral Session 4
15.00 – 15.15	Convolutional Vision Transformer Modeling for Spectrogram Image Processing in the Detection of North Atlantic Right Whales Up-Call
15.15 – 15.30	An Analysis of Synthetic Data for Improving Performance of Skeleton-Based Fall down Detection Models
15.30 – 15.45	Elevating Air Quality Forecasting: Integrating Hybrid Clustering Techniques with Long Short-Term Memory Networks
15.45 – 16.00	Transfer Learning Approach for Rainfall Class Amount Prediction Using Uganda's Lake Victoria Basin Weather Dataset

Program Schedule

Day 3 Sunday August 25, 2024

TIME	EVENT
09.00 - 11.45	Oral Session 5
09.00 – 09.15	Addressing Gender Bias: A Fundamental Approach to AI in Mental Health
09.15 – 09.30	Modification of Sand Cat Swarm Optimization for Classification Problems
09.30 – 09.45	Grapevine Leaf Disease Classification Using Deep Convolutional Neural Networks
09.45 – 10.00	Effect of Sliding Window Sizes on Sensor-Based Human Activity Recognition Using Smartwatch Sensors and Deep Learning Approaches
10.00 – 10.15	An Improvement on Exploration Step of Whale Optimization Algorithm with Levy Distribution for Classification Problems
10.15 – 10.30	Integrating In-Ear Wearable Sensors with Deep Learning for Head and Facial Movement Analysis
10.30 – 10.45	Evaluating GRNN, Decision Tree, and Random Forest: A Gas Turbine Emission Prediction Comparative Study
10.45 – 11.00	Ensemble Deep Learning Network for Enhancing Performances of Sensor-Based Physical Activity Recognition Based on IMU Sensor Data
11.00 – 11.15	Hybrid PSO-CNN Model for Cross-Domain Adaptation Sentiment Analysis
11.15 – 11.30	Design and Development of a Vertical Garden Station with Plants and an Automatic Fogging System for PM2.5 Reduction
11.30 – 11.45	Data Veracity Analysis in Social Medias – A Review

turnitin Page 30 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Web URLs Phishing Detection Model with Random Forest Algorithm

1st Aulia Kharisma Putri
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
aulia.putri@student.umn.ac.id

2nd Jansen Wiratama
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

3rd Samuel Ady Sanjaya
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

4th Santo Fernandi Wijaya
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
santo.fernandi@umn.ac.id

5th Monika Evelin Johan
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
monika.evelin@umn.ac.id

6th Ahmad Faza
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
ahmad.faza@umn.ac.id

***Abstract*— As internet users grow and technology evolves, so do the security risks, one example being phishing. Phishing is an attempt to obtain important information from someone, such as username, password, and other sensitive data, by providing a fake website that resembles the original. This research focuses on the problem of phishing website URLs that are increasing in number. Creating a model using an Algorithm that can detect phishing website URLs. The classification algorithm model that will be used in this research is Random Forest, which will be evaluated based on the confusion matrix value. The accuracy result is 99%. Second, the f1 score test result is 99.1%. The third result of recall testing is 99.3%. The last test result is a precision of 98.9%. With high accuracy, f1 score, recall, and precision values, the model created using the Random Forest algorithm can be applied well to applications in web URLs, phishing detecting, analyzing fake URL patterns, and identifying suspected links as fake web URLs.**

Keywords*— *CRISP-DM, phishing detection, random forest, web URLs

I. Introduction

Information technology is developing rapidly along with the times, especially in the field of Technology and Information. Currently, it has experienced rapid progress and provides various benefits. This rapid improvement of technology makes almost all activities easier, especially using the internet, especially through website media [1]. On the other hand, this convenience can be an opportunity for security risks for those who are still not familiar with online transaction security procedures. The security risk can be exploited by internet criminals to obtain confidential information, such as personal data, e-mail passwords, and even financial information, such as credit card data and online banking accounts, without the internet user's knowledge [2].

As internet users grow and technology develops, risks to security are increasingly diverse. One example is the practice of phishing. Phishing itself is an attempt to obtain crucial information from an individual, such as usernames, passwords, and other sensitive data, by providing a fake website that resembles the original [3]. Phishing websites will be carefully designed by internet criminals to resemble genuine sites, including appearance, content, domain URLs, and other elements, with the aim of deceiving victims (internet users). The main goal is to make victims believe that they are accessing a legitimate website page.

The impact of phishing can include financial losses and data loss and cause significant losses to victims. Therefore, it is necessary to detect links or uniform resource locators (URLs). Phishing detection is a way to find out whether a website URL address is fake or real. There are several ways to assess how safe a URL is, such as using blacklists, allowlists, statistics, or machine learning technology. Among all, machine learning technology is better because it is more efficient and accurate. This technology uses a special algorithm model to understand dangerous URL patterns and is able to detect the type of URL, whether it is a phishing or a safe site, according to needs [4].

HTML content analysis allows monitoring suspicious changes in the structure of web pages, while domain authority evaluation provides an idea of the legitimacy of the information source. Additionally, observation of script code helps in detecting suspicious activity or manipulation attempts that could harm users. By combining these three aspects, data mining models help create a strong layer of protection to identify and prevent URL-based phishing attacks efficiently [5].

Some of the functions of data mining include analysis of associations between data, data classification, data clustering, prediction, and others. This research uses a data mining-based model that aims to increase accuracy in classifying phishing URLs in the context of web security. By utilizing a data mining model, this model is designed to accurately detect and classify URLs that have the characteristics of phishing attacks. Data classification is a process of finding a model or function that can explain and differentiate data classes and concepts [6]. In this research, the functionality applied is data classification and uses a Random Forest Algorithm.

II. Methodology

Based on the dataset obtained, the method for solving the problem was determined using the Random Forest (RF), Support Machine Vector(SVM), and K-Nearest Neighbor (KNN) algorithms to determine the most accurate results. The determination of this method or algorithm is based on relevant literature studies and adapted to the needs and characteristics of existing data [7], [8]. After determining the approach, the data processing process begins with steps organized based on the CRISP-DM framework. CRISP-DM (Cross Industry Standard Process for Data Mining) is a framework that is widely used in various industries to carry out data science processes. This methodology details the stages and tasks

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

involved in a data science project and explains the relationship between each task [7].

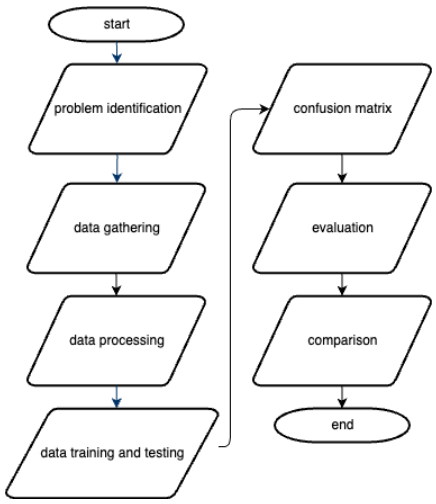


Fig 1. Research Flow

A. Business Understanding

The first stage of this research focuses on analyzing phishing cases in Indonesia and researching the corresponding data. Business Understanding in phishing web URL detection includes an in-depth understanding of web security threats, organizational needs regarding online security, and the impact of phishing attacks. The goal is to identify key aspects related to preventing, detecting, and protecting against phishing attacks on web URLs. This research aims to detect phishing website URLs using classification methods such as Random Forest, K-Nearest Neighbors, and Support Vector Machine.

Random Forest is a learning algorithm that is known for its ability to produce accurate predictions, especially in a framework with high-dimensional data [9]. The main advantage of Random Forest is its ability to overcome the overfitting problem that often occurs in complex models and its ability to perform well without requiring very specific parameter tuning [10]. This advantage makes Random Forest a popular choice in data analysis, especially in situations where high prediction accuracy is required without sacrificing model complexity or requiring complicated parameter tuning [11].

Support Vector Machine (SVM) is a machine learning algorithm used to create classification and regression models. The goal is to find the best-dividing line between two classes of data [12]. This algorithm has key parameters such as kernel parameters that allow data transformation into a higher dimensional space to increase class separation, as well as penalty parameters that regulate the trade-off between classification error on the training data and the separation margin between classes [13].

K-Nearest Neighbors is an algorithm in machine learning that classifies data based on similarity to previously existing training data [14]. This KNN method utilizes the principle that new data will be placed in the same class as the majority of classes of its nearest neighbors in feature space. This method is often used for solving classification problems, where data will be placed in categories based on the majority of categories from its nearest neighbors in feature space [15].

B. Data Understanding

The second stage of this research focuses on retrieving data from the Kaggle platform. Kaggle is a data science platform that offers access to daily and weekly time series that include exogenous variables as well as business hierarchy information [16]. This process starts with automation on the platform to get listing links and carry out the scraping process. Kaggle was chosen as the data source because the datasets there have undergone curation and pre-processing, ensuring data quality and integrity.

TABLE I. SAMPLE OF PHISHING DATASET

Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting
0	1	1	1	1	1
1	1	1	0	1	1
2	1	0	1	1	1
3	1	0	-1	1	1
4	-1	0	-1	1	-1
5	1	0	-1	1	1
6	1	0	1	1	1
7	1	0	-1	1	1
8	1	1	-1	1	1
9	1	1	1	1	1
10	1	1	-1	1	1
11	-1	1	-1	1	-1
12	1	1	-1	1	1
13	1	1	-1	1	1
14	1	-1	-1	-1	1
15	1	-1	-1	1	1
16	1	-1	1	1	1

The result is a dataset with 34 attributes, including 32 attributes that can be used for analysis such as Class, Using IP, Long URL, Short URL, Symbol@, Redirecting, Prefix Suffix, Sub Domains, HTTPS, Domain Reg Len, Favicon, Non-Std Port, HTTPS Domain URL, Request URL, Anchor URL, Links In Script Tags, Server From Handler, Email Info, Abnormal URL, Website Forwarding, Status Bar Cust, Disable Right Click, Using Popup Window, Iframe Redirection, Age of Domain, DNS Recording, Website Traffic, Page Rank, Google Index, Links Pointing to Page, Stats Report. The total amount of data in this dataset reaches more than 11,000 data.

C. Data Preparation

In the third phase of the research, the focus was on data cleaning and processing. Here is a detailed breakdown of this phase: Missing Values Check: The dataset was examined for missing or incomplete values. This step is crucial to ensure the integrity and quality of the data. The missing values were checked using the ISNA () function in the dataset, and the sum of missing values for each attribute was calculated and displayed. The output shows no missing values for any attributes, as indicated by the zeroes next to each attribute name.

Data Description: The data was described to determine its types and attributes. This step involves identifying whether the data is numerical or categorical, which helps decide the appropriate processing techniques. The attributes in the dataset were listed, showing a total of 32 attributes. These attributes are shown in Fig 2:

```
#cek missing value


missing_values = data.isna().sum()
print(missing_values)
```

```
Index                0
UsingIP              0
LongURL              0
ShortURL             0
Symbol@              0
Redirecting//        0
PrefixSuffix-        0
SubDomains           0
HTTPS                0
DomainRegLen         0
Favicon              0
NonStdPort           0
HTTPSDomainURL       0
RequestURL           0
AnchorURL            0
LinksInScriptTags    0
ServerFormHandler    0
InfoEmail            0
AbnormalURL          0
WebsiteForwarding    0
```

Fig 2. Missing Value Checking

The dataset comprises 32 attributes and more than 11,000 data points. The data preparation process was divided into two main parts are shown in Fig 3:

```
# Data Split
X = data_satu.drop(["class"],axis =1)
y = data_satu["class"]
```

Fig 3. Data Split

Data Cleansing involves removing any inconsistencies, errors, or irrelevant parts of the data. Given that no missing values were found, the data was already complete. Data Splitting involves dividing the dataset into subsets, typically for training and testing purposes in machine learning applications. Overall, the data preparation phase ensured that the dataset was ready for analysis and further processing, maintaining a high data quality and integrity standard.

D. Data Modelling

The fourth stage of this research includes modeling, which involves selecting the model and algorithm to be used, as well as implementing the algorithm. This research uses data modeling in the form of classification by displaying accuracy, precision, recall, and F1 score values. At this stage, the data model is adjusted as needed to achieve the desired results. In this research, three classification algorithms were chosen, namely Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The selection of the algorithm is based on evaluating the advantages, disadvantages, and information from previous research related to predicting phishing website URLs using data mining. The modeling stage was carried out using Google Collab tools. Google Collaboratory is a free cloud computing service provided by Google. The advantage of Google Colab lies in its ease of creating data visualizations because it does not require software installation on the user's computer [17].

E. Evaluation

The fifth stage of the research involved implementation using Python. Python, a programming language known for its ease of use, Python has gained immense popularity in both industrial and academic circles. The advantage of this language lies in its ability to be interpreted and executed by computers easily and to be accessed for free [18]. Python has a number of advantages and features that differentiate it from other programming languages [19]. This research uses the metrics A (Accuracy), P (Precision), R (Recall), and F1 Score to assess the performance of the classification model that has been built. Confusion matrix is a useful evaluation technique in the classification and prediction process [20].

Accuracy (A) describes how precise the algorithm model is in making accurate predictions. This accuracy is the proportion of correct predictions (both positive and negative) compared to the overall data [21].

$$\boldsymbol{A} = \frac{(\boldsymbol{TP} + \boldsymbol{FN})}{(\boldsymbol{TP} + \boldsymbol{FN} + \boldsymbol{FP} + \boldsymbol{FN}} \times \mathbf{100}\% \quad (1)$$

Precision (P) reflects the level of accuracy between the requested data and the model prediction results by comparing true positive predictions to the overall positive predicted results [22]. This precision aims to reduce the number of positive examples that are classified incorrectly.

$$\boldsymbol{P} = \frac{\boldsymbol{TP}}{\boldsymbol{FP} + \boldsymbol{TP}} \times \mathbf{100}\% \quad (2)$$

Recall (R), which is often referred to as sensitivity or true positive rate, evaluates the fraction of positive cases that are actually correctly identified by the model [21]. This recall becomes crucial when the cost of false negatives is high, as the focus is on reducing the number of missed positive examples.

$$\boldsymbol{R} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}} \times \mathbf{100}\% \quad (3)$$

F1 Score is a balanced measure that considers both precision and recall. This F1 Score is useful when dealing with imbalanced datasets, where one class may be more dominant than another [22]. The F1 score calculates the harmonic mean of precision and recall, providing a single value that balances between false positives (FP) and false negatives (FN).

$$\boldsymbol{F1} = \mathbf{2} * \frac{(\boldsymbol{recall} + \boldsymbol{precision})}{(\boldsymbol{recall} + \boldsymbol{precision})} \quad (4)$$

```
features = ['AnchorURL', 'HTTPS', 'WebsiteTraffic', 'LinksInScriptTags', 'SubDomains']
target = 'class'

# Pisahkan data menjadi data training dan data testing
X_train, X_test, y_train, y_test = train_test_split(df_satu[features], df_satu[target], test_size=0.2, random_state=42)

# Skala fitur-fitur untuk beberapa algoritma
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Fig 4. Data Modelling Evaluation

Evaluation is the assessment stage of the model that has been created at the modeling stage. The main focus is to ensure that the results obtained are in line with previously established business objectives. By using these metrics, this research can measure the extent to which the model can provide accurate and relevant predictions in identifying phishing website URLs.

III. Results and Discussion

After carrying out the stages in CRISP-DM, a confusion matrix value is produced to measure the evaluation performance of each machine learning model. The following are the resulting values:

```
SVM Metrics:
Accuracy: 0.9280868385345997
Precision: 0.9304
Recall: 0.9417004048582996
F1 Score: 0.9360160965794769

Random Forest Metrics:
Accuracy: 0.9371325192220714
Precision: 0.9328593996840442
Recall: 0.9562753036437247
F1 Score: 0.9444222311075571

KNN Metrics:
Accuracy: 0.9271822704658526
Precision: 0.9182242990654206
Recall: 0.9546558704453442
F1 Score: 0.9360857483128225
```

Fig 5. Confusion Matrix Results

The evaluation results that have been presented illustrate the relative performance of the three models in classifying data. Analysis of these values provides a comprehensive view of the strengths and weaknesses of each model in the context of phishing URL detection.

TABLE II. RESULTS OF ALGORITHMS PERFORMANCE

Algorithms	Accuracy	Precision	Recall	F1 Score
SVM	0.928	0.934	0.941	0.936
RF	0.937	0.932	0.956	0.944
KNN	0.927	0.918	0.954	0.936

In the Random Forest model, there is an increase in performance with an accuracy of 0.937, indicating the correct level of predictions from this model. A precision of 0.933 indicates the model's level of accuracy in classifying positive data, while recall reaches 0.956, indicating the model's ability to find positive data instances. The F1 score of 0.944 shows a balance between precision and recall in the Random Forest model.

IV. CONCLUSION

In this research, we have investigated important aspects of detecting phishing websites, exploring various machine learning techniques and their effectiveness in dealing with the ever-growing cybersecurity threat of phishing attacks. The increasing level of fraudulent activity has become a major challenge for individuals and organizations around the world, driving the need to develop robust and efficient methods for identifying and preventing these fraudulent websites.

This analysis confirms that the superiority of Random Forest depends not only on the overall availability of attributes but also on its ability to adapt to the most relevant attributes. This result analysis indicates that Random Forest has a better capacity to understand patterns in data in a more adaptive way than SVM and KNN. In the context of developing a model for web URL phishing detection, the selection of appropriate attributes plays a very important role. Although all algorithms show good potential, Random Forest's ability to adapt to the most significant information underlines the importance of selecting relevant features in the development of a reliable and effective phishing detection model. This model can be a basis for optimizing models that will be implemented in phishing detection practices in a wider environment.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Universitas Multimedia Nusantara for their invaluable support, which played a pivotal role in the successful completion of this research endeavor. Their substantial contribution was instrumental in achieving our objectives, and we are deeply grateful for their unwavering assistance.

REFERENCES

[1] P. Subarkah and A. N. Ikhsan, "Identifikasi Website Phishing Menggunakan Algoritma Classification And Regression Trees (CART)," *J. Ilm. Inform.*, vol. 6, no. 2, pp. 127–136, 2021, doi: 10.35316/jimi.v6i2.1342.

[2] N. B. Putri and A. W. Wijayanto, "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing," *Komputika J. Sist. Komput.*, vol. 11, no. 1, pp. 59–66, 2022, doi: 10.34010/komputika.v11i1.4350.

[3] A. N. I. Pungkas Subarkah, "' Aplikasi Pendeteksi Website Phishing Menggunakan Machine Learning ,'" *J. Ilm. Inform. Inform.*, 2020.

[4] S. Saxena, A. Shrivastava, and V. Birchha, "A Proposal on Phishing URL Classification for Web Security," *Int. J. Comput. Appl.*, vol. 178, no. 39, pp. 47–49, 2019, doi: 10.5120/ijca2019919282.

[5] F. Carroll, J. A. Adejobi, and R. Montasari, "How Good Are We at Detecting a Phishing Attack? Investigating the Evolving Phishing Attack Email and Why It Continues to Successfully Deceive Society.," *SN Comput. Sci.*, vol. 3, no. 2, p. 170, 2022, doi: 10.1007/s42979-022-01069-1.

[6] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A Hybrid Approach for Alluring Ads Phishing Attack Detection Using Machine Learning," *Sensors*, vol. 23, no. 19, pp. 1–27, 2023, doi: 10.3390/s23198070.

[7] D. A. Kristiyanti and S. Hardani, "Sentiment Analysis of Public Acceptance of Covid-19 Vaccines Types in Indonesia using Naïve Bayes, Support Vector Machine, and Long Short-Term Memory (LSTM)," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 3 SE-Information Systems Engineering Articles, Jun. 2023, doi: 10.29207/resti.v7i3.4737.

[8] M. H. Fadly and M. E. Johan, "Web-Based Heart Disease Prediction by Comparison and Implementation of SVM, AdaBoost, and Hybrid SVM-AdaBoost Algorithms," in *2023 7th International Conference on New Media Studies (CONMEDIA)*, 2023, pp. 257–262. doi: 10.1109/CONMEDIA60526.2023.10428512.

[9] H. Wang and G. Wang, "Improving random forest algorithm by Lasso method," *J. Stat. Comput. Simul.*, vol. 91, no. 2, pp. 353–367, 2021, doi: 10.1080/00949655.2020.1814776.

[10] W. Feng, C. Ma, G. Zhao, and R. Zhang, "FSRF:An Improved Random Forest for Classification," *Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020*, pp. 173–178, 2020, doi: 10.1109/AEECA49918.2020.9213456.

[11] C. Slimani, "RaFIO : A Random Forest I / O-Aware algorithm," *SIGAPP Symp. Appl. Comput.*, no. 1, pp. 521–528, 2021.

[12] E. Ameisen and an O. M. C. Safari, *Building Machine Learning Powered Applications*. 2020.

[13] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, no. xxxx, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.

[14] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification,"

2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019, no. Icicccs, pp. 1255–1260, 2019, doi: 10.1109/ICCS45141.2019.9065747.

[15] S. Zhang, "Challenges in KNN Classification," *IEEE Trans. Knowl. Data Eng.*, pp. 1–13, 2021, doi: 10.1109/TKDE.2021.3049250.

[16] C. S. Bojer and J. P. Meldgaard, "Kaggle forecasting competitions: An overlooked learning opportunity," *Int. J. Forecast.*, vol. 37, no. 2, pp. 587–603, 2021, doi: 10.1016/j.ijforecast.2020.07.007.

[17] R. Gelar Guntara, "Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab," *J. Ilm. Multidisiplin*, vol. 2, no. 6, pp. 2091–2100, 2023.

[18] Muhammad Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma," *JTIM J. Tek. Inform. Mahakarya*, vol. 03, no. 2, pp. 37–44, 2020.

[19] V. Thangarajah, "Python current trend applications-an overview," Oct. 2019.

[20] R. Ridho and H. Hendra, "Klasifikasi Diagnosis Penyakit Covid-19 Menggunakan Metode Decision Tree," *JUST IT J. Sist. Informasi, Teknol. Inf. dan Komput.*, vol. 11, no. 3, pp. 69–75, 2022.

[21] I. Sapuan, M. H. Fauzan, and C. Juliane, "Implementasi Data Mining untuk Klasterisasi dan Prediksi Kelompok Keluarga," *JTERA (Jurnal Teknol. Rekayasa)*, vol. 7, no. 1, p. 149, 2022, doi: 10.31544/jtera.v7.i1.2022.149-156.

[22] K. Omari, "Comparative Study of Machine Learning Algorithms for Phishing Website Detection," *IJACS*, vol. 14, No 9, 2023, doi: 10.14569/IJACSA.2023.0140945.

57 turnitin Page 35 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Convolutional Vision Transformer Modeling for Spectrogram Image Processing in the Detection of North Atlantic Right Whales Up-Call

1st Siti Ummi Masruroh
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
ummi.masruroh@uinjkt.ac.id

2nd Muhammad Destamal Junas
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
muhammad.junas19@mhs.uinjkt.ac.id

3rd Khodijah Hulliyah
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
khodijah.hulliyah@uinjkt.ac.id

4th Husni Teja Sukmana
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
husniteja@uinjkt.ac.id

5th Rizka Amalia Putri
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
rizka.amalia18@mhs.uinjkt.ac.id

6th Saepul Aripiyanto
Department of Informatics
UIN Syarif Hidayatullah
Jakarta, Indonesia
saepul.aripiyanto@uinjkt.ac.id

***Abstract*—Whale sound detection is an important topic in marine conservation and research. The North Atlantic Right Whale is a whale species whose population is on the brink of extinction. Some of the threats to the North Atlantic Right Whale population at this time are fishing nets or gear that can get caught on whales and whale collisions with ships. To be able to reduce these events, it is necessary to determine the whale hotspot area to be avoided by ships. In this case, the researcher tries to use a spectrogram image processing approach with the Convolutional Vision Transformer model of underwater sound recordings with whale calls as the main parameter in whale detection. The use of the Convolutional Vision Transformer model is a new approach to spectrogram image processing. The Convolutional Vision Transformer model uses convolutional layers combined with Transformer blocks which in theory can speed up the training process. The model can process spectrogram images well, with 97.25% accuracy, 97.26% F1 Score, 97.34% Precision, and 97.25% Recall. With a faster training runtime compared to the Vision Transformer model.**

Keywords—Convolutional Vision Transformer, Deep Learning, Transformer, North Atlantic Right Whale, Whale Call

I. INTRODUCTION

11 Whale sound detection is an important topic in the field of conservation and marine research. Whale sounds, known as whale songs, have specific patterns and characteristics that can be used to identify whale species, study their behavior, and monitor whale populations in the ocean [1]. Apart from that, whale sound detection can be implemented in the shipping industry, so that we can determine the best transport routes to reduce the impact caused by ships on the marine environment.

In this study, researchers detected one type of whale, namely the North Atlantic Right Whale. Based on a report compiled by the World Wildlife Fund WWF, the North Atlantic Right Whale is a type of whale whose population is almost extinct. This population decline was initially caused by whaling which became widespread in the 1890s. Whale hunting is no longer a problem today, but the North Atlantic Right Whale population has not recovered to its previous level In this study, researchers detected one type of whale, namely the North Atlantic Right Whale. Based on a report compiled by the World Wildlife Fund WWF, the North Atlantic Right Whale is a type of whale whose population is almost extinct. This population decline was initially caused by whaling which became widespread in the 1890s. Whale hunting is no longer a problem today, but the North Atlantic Right Whale population has not recovered to its previous level.

Several things pose a threat to the population of North Atlantic Right Whales At this time it is nets or fishing equipment that can get caught in whales and whales collide with ships. The increasing amount of underwater noise pollution produced by humans also affects whales, because these sounds can disrupt communication between whales and increase stress in whales [2]. To preserve this type of whale, NOAA or the National Oceanic and Atmospheric Administration places limits on ship speed in several sea areas to reduce collisions between whales and ships. These regulations are very important in the conservation process because the existence of these regulations can reduce the risk of death or injury to whales. After all, ships of any size can injure whales that are on the surface of the sea.

The areas regulated by these regulations are hot areas or hot spots where the whale population is located [3]. Therefore, an appropriate algorithm is needed to be able to determine these hot areas. In this case, the researchers tried to use a spectrogram image processing approach from underwater sound recordings with the sound of whale songs as the main parameter in whale detection.

Detection of whale sounds through spectrogram analysis has several challenges. First, whale sound spectrograms often have complex patterns and differ between whale species [4]. This makes detecting whale sounds a difficult task. Additionally, whale sounds are often drowned out by ocean noise, such as the sound of waves, ships, or other marine animals. This noise can interfere with the detection process and produce inaccurate results. Therefore, an effective method is needed to overcome this challenge.

The use of the Convolutional Vision Transformer (CvT) for whale sound detection through spectrogram analysis has high relevance in today's industry [5]. In the field of conservation and marine research, automatic and accurate detection of whale sounds is critical to understanding whale populations, their behavior, and the impact of environmental change on them. This automated method can help reduce the time and effort required for whale sound analysis, as well as increase accuracy and efficiency in data collection. In addition, the use of CVT as a new approach to whale sound

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

detection can also contribute to the development of sound recognition and signal processing technology in the audio and communications industry.

Based on research conducted by Dosovitsky [5]. it can be concluded that Vision Transformer can produce state-of-the-art performance in image processing. However, the drawback of Vision Transformer is that the process requires quite a large number of parameters in the pre-training process so that the model can get good performance. To overcome this, the Convolutional Vision Transformer was developed to obtain performance like the Vision Transformer but reduces the parameters required in the pre-training process by adding convolution blocks to the transformer model so that model performance is maintained. With data obtained from previous research [6].

Therefore, the author tries to utilize the Convolutional Vision Transformer model for processing spectrogram images in the case study of North Atlantic Right Whale detection.

II. Literature Review

A. Threats to the North Atlantic Right Whale Population

Several threats to the North Atlantic Right Whale population currently include entanglement in fishing gear or equipment that can ensnare whales, as well as collisions between whales and ships. The increasing amount of underwater noise pollution generated by humans also affects whales, as these sounds can disrupt inter-whale communication and elevate stress levels among whales [7].

In the preservation efforts of this whale species, the NOAA or National Oceanic and Atmospheric Administration has implemented ship speed restrictions in certain marine areas to mitigate collisions between whales and ships. Such regulations are crucial in conservation processes as they can reduce the risk of mortality or injury to whales, given that vessels of any size can harm whales when they surface. The regulated areas encompass hotspots where whale populations are concentrated [8]. Hence, an appropriate algorithm is needed to identify these hotspots. In this context, researchers are attempting to utilize a spectrogram image processing approach from underwater sound recordings, with whale vocalizations as the primary parameter for whale detection.

B. Spectrogram

In this research, spectrograms aid in identifying sound patterns, harmonics, and changes in audio signals used in speech recognition, music analysis, and sound effects design. Additionally, in the medical field, spectrograms are also applied to medical imaging, such as analyzing ultrasound signals to visualize the frequency content of tissues. In radar and sonar systems, spectrograms assist in target identification and understanding the characteristics of signals reflected from objects. Even in mechanical and structural engineering, spectrograms are used to analyze vibrations and identify resonance frequencies [9].

C. Convolutional Vision Transformer

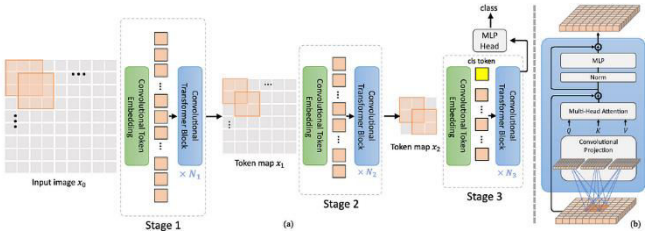


Fig. 1. CVT Model Architecture

The utilization of the Convolutional Vision Transformer (CvT) for whale sound detection through spectrogram analysis holds high relevance in the current industry landscape. In the fields of conservation and marine research, automated and accurate whale sound detection is crucial for understanding whale populations, their behaviors, and the impacts of environmental changes on them. This automated method can help reduce the time and effort required for whale sound analysis while enhancing accuracy and efficiency in data collection. Moreover, employing CvT as a novel approach in whale sound detection can also contribute to the development of sound recognition technology and signal processing in the audio and communication industries.

The Convolutional Vision Transformer (CVT) model differs from the Vision Transformer model in its architectural flow. The CVT model features a multi-stage architecture adapted from CNN models where CVT consists of 3 stages, and each stage comprises 2 parts. CNN is a sort of feedforward neural network that CNNs utilize to extract features from data. Like any other neural network model, CNNs are built on layers of neurons inside the organization, which allows them to learn hierarchical representations. The neurons in the layers are linked using weights and biases [10].

D. Convolutional Token Embedding

The convolution process in this block in the CvT model aims to understand local spatial contexts from low-level to semantic primitives. This block receives a two-dimensional image input or token map from a two-dimensional image that has passed through the reshaping stage in the previous step, denoted as f(x_i-1) [6].

E. Convolutional Projection

The main objective of convolutional projection is to understand the local spatial context as a means for the model to discern patterns among adjacent pixels in the image. Additionally, convolutional projection can provide computational efficiency by implementing undersampling in the Key (K) and Value (V) matrices during certain computations. Therefore, this model can become more efficient in terms of processing time and resource utilization without significantly affecting model quality. In the CvT model, the Multi-Head Self-Attention block is substituted with depth-wise convolution, yielding convolutional projection.

F. Whale Call Detection

In the marine environment, the penetration of light into water is highly limited, thus visual cues for marine organisms are also restricted. Therefore, cetaceans such as whales and dolphins rely heavily on sound in every aspect of their lives, with each species producing specific types of vocalizations. Many cetacean vocalizations can be identified to the species

or even population level, allowing the use of acoustic recording tools to detect their presence. With advancements in acoustic recording technology and the decreasing cost of equipment, there has been a proliferation of high-resolution acoustic data. With this abundance of data, the challenge lies not in acquiring the data but rather in processing it. Consequently, scientists have begun exploring ways to speed up recognizing whale calls or whale sounds through automation using machine learning [11].

III. Method

The type of research carried out is quantitative research, which is a type of research that uses numbers in processing data to produce structured information. Based on this understanding, quantitative research is very relevant to the research that will be carried out because for machine learning modeling data processing is carried out to obtain accurate information from the CvT model for detecting NARW whale songs. In this research, the author uses the following machine learning model development method:

A. Data Collection

The data acquisition process carried out by researchers was taking data from the Kaggle site entitled "Whale Detection Challenge: 255x255 png Dataset" which contains 30,000 spectrogram images based on underwater audio recordings recorded by The Marinexplore and Cornell University. Of the 30,000 images, the data is divided into two labels, namely whale and nowhale with a total of 22,973 images for nowhale labels, and 7,027 images for whale labels. However, because the data sharing ratio, according to the author, is too far, it is feared that there may be invisible patterns from fewer data labels. Therefore, the researchers reduced the data from 30,000 data to 20,000 image data by dividing 13,000 images with the no whale label and 7000 images with the whale label.

B. Data Pre-Processing

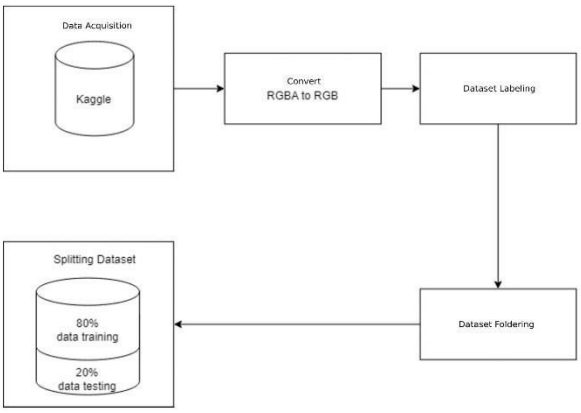


Fig. 2. Data Pre-Processing Stages

Data pre-processing is an important step in a data analysis project which aims to extract important features from the data and eliminate noise or irrelevant data [12].

Because the number of channels in the data that researchers use does not follow the provisions of the model that will be used, the channels in the data are still RGBA or Red, Green, Blue, and Alpha, whereas, in the CvT-13 model, the data that will be processed must have three channels, namely RGB or Red, Green, and Blue. Therefore, the researcher created a script to change the channel from RGBA to RGB. After the conversion, the converted data can be processed to go through the labeling stage based on the Excel file provided by The Marinexplore and Cornell University. After labeling, the data will be folderized according to the provisions in the documentation on HuggingFace so that it can be saved in dataset format. After the folderization process, the dataset can be imported using the load_dataset function. After the dataset has been processed, the dataset can be divided into 80% training data and 20% testing or validation data. If the data has gone through several pre-processing stages, the dataset can be entered into the model for the next stage of training.

C. Model Training

Model Training or model training is a process in machine learning and artificial intelligence, where the machine learning algorithm is given an input dataset to be able to generalize patterns from that data. After model training is complete, the trained model can be used to make predictions on data that has never been seen before. Regular model updating and retraining is often necessary to maintain model accuracy and relevance.

The model training process consists of several iterations, during which the model gradually adapts to patterns in the training data. In each iteration, the model updates its parameters based on the difference between the predictions produced by the model and the actual labels in the training data. This allows the model to learn and adapt over time.

However, it is important to remember that pre-trained models are not always perfect. Therefore, regular updating and retraining of the model is often necessary. New data received, changes in the environment, or increases in computing resources may cause the model to become outdated or inaccurate.

Therefore, to maintain optimal levels of accuracy and relevance, models need to be regularly updated and improved through a retraining process. In the training process, researchers experimented by changing several training arguments, namely learning rate and weight decay. The selected values for the learning rate are 1e-5, 2e-5, and 3e-5. Meanwhile, for weight decay, the values are 0.1, 0.01, and 0.001.

IV. Results and Discussion

Before implementation, researchers looked for several literature study sources as references and comparisons in the model implementation process. A summary of the research results is as follows Authors and Affiliations

TABLE I. Literature Review Study

Research	Conclusion
Detection of North Atlantic Right Whales with A Hybrid System of CNN and Dictionary Learning	The use of CNN and DL hybrid systems can reach quite good accuracy, by using CNN as a feature extractor and replacing softmax with Dictionary Learning proven to increase accuracy.
A Convolutional Neural Network for Automated Detection of Humpback Whale Song in a Diverse, Long-Term Passive Acoustic Dataset [1]	Using a hybrid CNN and DL system can achieve fairly good accuracy, by using CNN as a feature extractor and replacing softmax with dictionary learning, it is proven to increase accuracy.

North Atlantic Right Whales Up-call Detection Using Multimodel Deep Learning [13]	Processing acoustic data with CNN can be an option if existing data is limited. CNNs can achieve excellent performance with little data.
North Atlantic Right Whale Call Detection with Convolutional Neural Networks [14]	CNN combined with SAE or stacked autoencoder can easily understand the context of data locally and globally. Relabeling also affects accuracy.
Right Whale Detection Using Artificial Neural Network and Principal Component Analysis [15]	Fully connected CNN can achieve good accuracy results in detecting NARW whale songs.

The initial stage that the author carried out was data acquisition according to the problems required for the needs of the training and model evaluation stages. The authors collected a dataset containing spectrogram images of underwater recordings, with a difference between recordings containing whale sounds and recordings that did not contain whale sounds.

After searching the internet, the author found a dataset on the Kaggle website with the title "Whale Detection Challenge: 255x255 png Dataset" which contains 30,000 spectrogram images from underwater recordings. With a total of 30,000 images, this dataset provides significant diversity in terms of the variety of underwater recordings. This allows the model to study different characteristics of whale sounds from different viewpoints and conditions. As a first step, authors need to check and validate the dataset to ensure its quality and consistency before using it in the model training and evaluation stages.

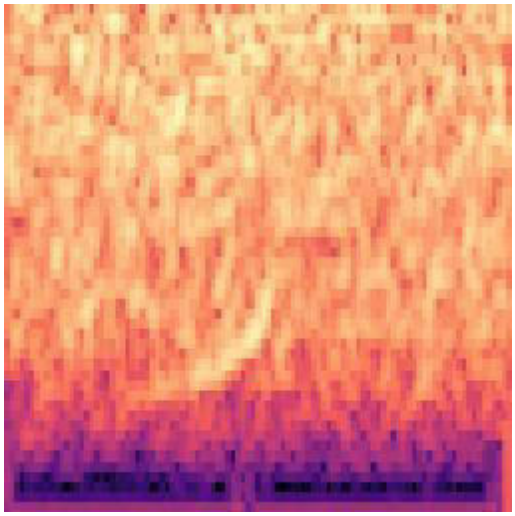


Fig. 3. Examples of Images On the Dataset

The spectrogram image is a visualization of a two-second underwater audio recording containing whale sounds. According to research conducted by spectrograms produced visualizes recordings in the range 0 Hz to 500 Hz. For example, the following is a spectrogram image that was used as an example data by Spaulding et al [16].

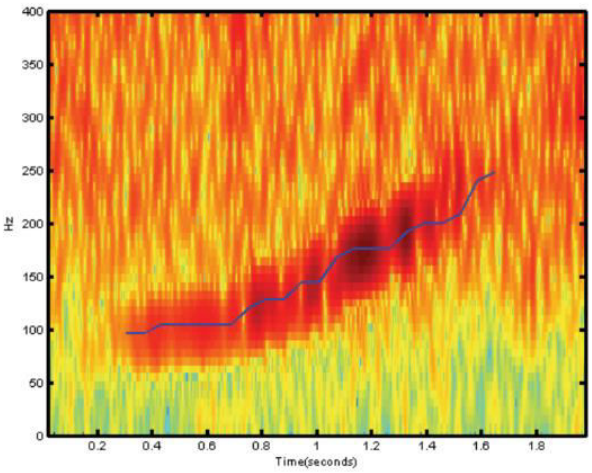


Fig. 4. Picture of Spectogram

It can be seen in the spectrogram graph above that whale songs experience an increase in strength at frequencies from 100 Hz to around 250 Hz to 300 Hz with the greatest strength around 200 Hz to 250 Hz. With the following basis, researchers will try to see the performance of the Convolutional Vision Transformer model to recognize features in images that have whale song characteristics that indicate the presence of NARW whales.

From the results of the dataset examination, the author found that the images in the relevant dataset did not have labels, so manual labeling was necessary. After conducting further searches, the author found the audio dataset used by the related dataset which was used as a spectrogram. In the audio dataset that the author found, there are image data labels in .csv form. Based on this, the author carried out labeling using CSV data sources. provided by the dataset creator.

The dataset then goes through several stages of data pre-processing. The processed data will be split into training data and testing data with a specified ratio. Next, the image data will go through an augmentation stage to increase the dataset with existing data. The stage that the researcher carried out was the use of libraryAutoImageProcessor to perform augmentation by config or rules applied to the model cvt-13. Then, researchers carry out transformations on the image data so that the image data can be processed in tensor form with functionsToTensor.

The images in the dataset that have been transformed will be included in the modelCvT-13 for model training. For the parameters carried out in the experiments, the researchers changed the parameters of weight decay as well as learning rate. The results of model training can be seen in the following image.

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.489700	0.275342	0.894500	0.892951	0.894788	0.894500
2	0.390100	0.200908	0.943000	0.943197	0.943681	0.943000
3	0.343500	0.161781	0.950500	0.950713	0.951388	0.950500
4	0.331400	0.163662	0.951833	0.951993	0.952420	0.951833
5	0.339400	0.145355	0.957500	0.957671	0.958250	0.957500

Fig. 5. Training Model Results

Apart from looking at the test table, we can see the model performance in the form of a confusion matrix as follows:

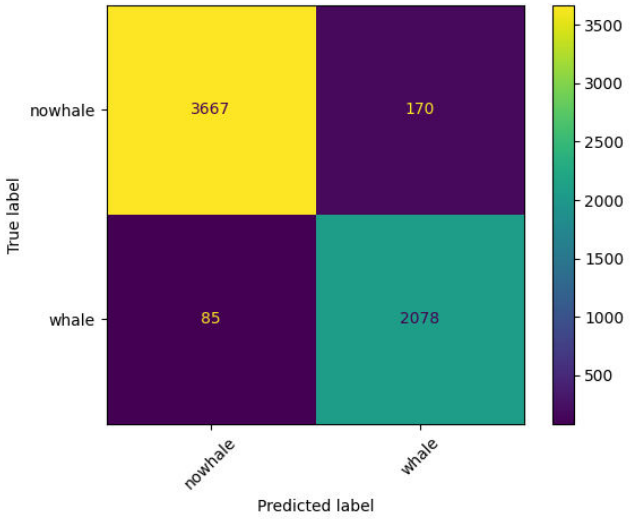


Fig. 6. Confusion Matrix

From a total of 6000 validation images, the model testing results succeeded in predicting 2078 positive images from a total of 2248 positive image data. Of the 3752 negative images, the model succeeded in predicting 3667 negative or True Negative images. Researchers tried to experiment to change the ratio of training data and testing data, which was initially 30% testing data and 70% training data, to 20% testing data and 80% training data. With the same hyperparameter argument, researchers obtained an increase in accuracy with the following details.

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.437500	0.228351	0.906250	0.904575	0.907783	0.906250
2	0.378500	0.141423	0.960750	0.960999	0.962340	0.960750
3	0.352900	0.126755	0.960000	0.960327	0.962580	0.960000
4	0.349800	0.109224	0.969500	0.969665	0.970609	0.969500
5	0.339600	0.108446	0.968250	0.968399	0.969124	0.968250

Fig. 7. Model Training Results With a Data Ratio of 20:80

The change in the ratio of testing data and training data has had quite a good impact, because there is an increase in the accuracy, F1, precision, and recall scores of the model. Based on the results of observations, researchers also did not find any significant increase in training time, because previously the model went through a training process of 1681 seconds, whereas with a ratio of 80:20, the model only experienced an increase in runtime of approximately 151 seconds.

Based on the training results and validation with test data with new dataset ratios, researchers tried to change several hyperparameters in the CVT model training process. The hyperparameters that will be experimented with are learning_rate and weight_decay. Learning rate is an important parameter in the model training stage. The learning rate controls how fast or slow a model learns during the training process and determines the step size when weights and biases are updated to minimize the loss function. Here the researcher tried to change the learning rate values to 1e-5, 2e-5, and 3e-5 with details of the experimental results in the following table.

TABLE II. LEARNING RATE COMPARISON TABLE

Learning rate	Accuracy	F1-Score	Precision	Recall	Runtime
1e-5	97.10%	97.11%	97.18%	97.10%	1768 s
2e-5	96.95%	96.96%	97%	96.95%	1796 s
3e-5	97.25%	97.26%	97.34%	97.25%	1752 s

As the researcher mentioned previously, apart from the learning rate, the researcher also tried to replace the value of the weight decay parameter with the following values, namely, 0.1, 0.01, and 0.001, with the following model training results.

TABLE III. WEIGHT DECAY COMPARISON TABLE

Weight Decay	Accuracy	F1-Score	Precision	Recall	Training Runtime
0.1	97.10%	97.11%	97.18%	97.10%	1768 seconds
0.01	97%	97%	97.10%	97%	2386 seconds
0.001	97.20%	97.20%	97.28%	97.20%	2400 sec

It can be seen in the table that decreasing weight decay does not significantly affect accuracy, f1 score, precision, and recall, but the smaller the weight decay, the longer the runtime in the training process. At the smallest weight decay, namely 0.001, the training runtime increased by 632 seconds or 10 minutes 32 seconds.

Based on data from training with the Vision Transformer model as a comparison, the accuracy of the Vision Transformer model is not that far from the accuracy of the CVT model. However, in the training process of the Vision Transformer model, the model experienced slight overfitting in the 4th and 5th epochs which we can observe, namely the training loss is lower than the validation loss. The training process for the Vision Transformer model also takes longer than the CVT model in the following table.

TABLE IV. COMPARISON BETWEEN CVT AND VIT

Model	Runtime	Samples/s	Steps/s	Total_flos	Train_loss
CvT	1752.063	45.66	4.163	1.417	0.318
YEAR	3313.708	21.124	2.112	5.424	0.054

In the model training stage, model performance is assessed based on several metrics that the researchers have displayed in the table above. Train_loss is a metric that measures the model's ability to minimize prediction errors during training. We can see that the CVT model has a higher train loss (0.318) compared to the ViT model (0.054). This shows that during the training stage, the CVT model has a greater error in predicting images in the training dataset.

Efficiency in the validation stage is also an important highlight. The CVT model shows significant efficiency, with a shorter execution time (40.20 seconds) and a much higher number of samples per second (samples per second) as well as a much higher number of steps per second (samples per second) compared to the ViT model. This efficiency indicates that the CVT model can process validation data more quickly

and efficiently, an aspect that can provide advantages in implementations that require fast prediction results.

The CVT model has better performance in terms of accuracy, F1-score, precision, and recall in testing data as well as advantages in faster data processing and inference times.

Apart from making comparisons with the ViT model, the author will also compare the prediction results from the CVT model with several previous studies. In several previous studies, the benchmarks for accuracy in detecting NARW whale songs were Up-Call Detection, non-up-call detection, and False Alarm.

Model	TP	FN	FP	TN	Up-call	Non-Up-call	False Alarm
CvT	1421	22	100	2457	98.47%	96.08%	3.2%
MFCC	172	36	61	1231	73.68%	97.86%	2.48%
DWT *single stage* +MFCC	181	52	34	1233	91.85%	97.32%	2.68%
DWT+MFCC +Linear SVM	215	1202	18	65	92.27%	94.87%	1.48%
TFP-2 features+LDA	187	1206	46	61	77.68%	97.32%	4.81%
Spectrogram+CNN+DL	-	-	-	-	92.37%	97.37%	1.42%

Fig. 8. Comparison with Previous Research

Based on this comparison, it can be seen that the CVT model can compete with previous models that researchers used from research [12] for the Spectrogram + CNN + DL model and also several models such as MFCC, DWT single stage + MFCC, DWT + MFCC + Linear SVM, and TFP-2 Features+LDA from research [16]. The CVT model got the highest Call Detection Rate, namely 98.47%, but the Non-Up Call Detection Rate it only 96.08%, 1.68% lower than the highest accuracy, namely the MFCC model with an accuracy of 97.86%. For False Alarm, CVT gets an accuracy of 3.91%, which is quite high compared to previous models. In the previous research that the author used as a comparison, there was no additional information regarding model performance during training so the author could not compare the efficiency of the model in the training process. The CVT model can compete with hybrid models in acoustic data processing. However, further research is needed to be able to optimize the CvT model to handle acoustic data cases. Please note, some of the models compared here do not use the same data sharing ratio, and some models use different preprocessing methods.

V. Conclusion

The research results show that the Convolutional Vision Transformer model, which has gone through a fine-tuning process, succeeded in detecting whales through spectrogram images with a high accuracy of 97.25%, F1-Score 97.26%, precision of 97.34%, and recall of 97.25% in five iterations of training data so that outperforms the regular Vision Transformer model. Apart from that, this model also shows higher efficiency in the training and validation data prediction process, with a training time of only 1752 seconds and a validation data prediction time of 40,201 seconds. Furthermore, this model requires less computing power, with total_flos of only 1,417 compared to 5,424 in the Vision Transformer model without convolution. After hyperparameter tuning, the model with the highest performance has a 5 epoch configuration, learning rate 3e-5, weight decay 0.001, and batch size 4 for training and evaluation per device.

Acknowledgment

A portion of this work is supported by Syarif Hidayatullah State Islamic University Jakarta's Center for Research and Publication. The reviewers' insightful critiques and recommendations, which enhanced the presentation, are also gratefully acknowledged by the writers.

References

[1] A. N. Allen *et al.*, "A Convolutional Neural Network for Automated Detection of Humpback Whale Song in a Diverse, Long-Term Passive Acoustic Dataset," *Front. Mar. Sci.*, vol. 8, 2021.

[2] C. Jhonson, "Protecting Blue Corridors," *WWF*, 2022. .

[3] "North Atlantic Right Whale | NOAA Fisheries." [Online]. Available: https://www.fisheries.noaa.gov/species/north-atlantic-right-whale. [Accessed: 15-Mar-2024].

[4] M. Thomas, B. Martin, K. Kowarski, B. Gaudet, and S. Matwin, "Marine Mammal Species Classification Using Convolutional Neural Networks and a Novel Acoustic Representation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11908 LNAI, pp. 290–305, 2020.

[5] A. Dosovitskiy *et al.*, "an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, 2021.

[6] H. Wu *et al.*, "CvT: Introducing Convolutions to Vision Transformers," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 22–31, 2021.

[7] C. Johnson *et al.*, "Protecting Blue."

[8] National Oceanic and Atmospheric Administration, "North Atlantic Right Whale," 2023. .

[9] S. Gupta, J. Jaafar, W. Fatimah, and A. Bansal, "FEATURE EXTRACTION USING MFCC," *Signal \& Image Process. An Int. J.*, vol. 4, pp. 101–108, 2013.

[10] S. U. Masruroh, M. F. Syahid, F. Munthaha, A. T. Muharram, and R. A. Putri, "Deep Convolutional Neural Networks Transfer Learning Comparison on Arabic Handwriting Recognition System," *Int. J. Informatics Vis.*, vol. 7, no. 2, pp. 330–337, 2023.

[11] C. Bergler, "ORCA-SPOT: An Automatic Killer Whale Sound Detection Toolkit Using Deep Learning," *Sci. Rep.*, vol. 9, 2019.

[12] S. Tang, S. Yuan, and Y. Zhu, "Data Preprocessing Techniques in Convolutional Neural Network Based on Fault Diagnosis towards Rotating Machinery," *IEEE Access*, vol. 8, pp. 149487–149496, 2020.

[13] A. K. Ibrahim, H. Zhuang, N. Erdol, and A. Muhmed Ali, "Detection of north atlantic right whales with a hybrid system of CNN and dictionary learning," *Proc. - 2018 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2018*, pp. 1210–1213, 2018.

[14] H. Glotin, C. Clark, Y. Lecun, P. Dugan, X. Halkias, and J. Sueur, *The 1st International Workshop on Machine Learning for Bioacoustics*, vol. 1, no. Icml. 2013.

[15] K. Pylypenko, "Right whale detection using artificial neural network and principal component analysis," *2015 IEEE 35th Int. Conf. Electron. Nanotechnology, ELNANO 2015 - Conf. Proc.*, pp. 370–373, 2015.

[16] E. Spaulding *et al.*, "An autonomous, near-real-time buoy system for automatic detection of North Atlantic right whale calls," *Proc. Meet. Acoust.*, vol. 6, no. 2009, 2009.

turnitin Page 41 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Data Veracity Analysis in Social Media – A review

Mohamed Amine BELLOUQI
Laboratoire Informatique et Système (LIS),
Faculty of Sciences, Hassan II University of Casablanca
Casablanca, Morocco
mohamedaminebellouqi@outlook.com

Ismail ASSAYAD
Laboratoire Informatique et Système (LIS),
Faculty of Sciences & ENSEM, Hassan II University of Casablanca
Casablanca, Morocco
iassayad@gmail.coms

***Abstract*—Due to their exponential growth, social media platforms have become the main source of information for many people. On these platforms, propaganda, fake news, and misinformation are widely disseminated, raising serious questions about the accuracy and veracity of the data. In this paper, we will cover a review of several key approaches in the context of data veracity analysis on social media, with a particular focus on rumor detection. By exploring these diverse strategies, the purpose of this paper is to contribute to the understanding of the multifaceted techniques applied in ensuring data veracity, specifically in the challenges of rumor detection on social media platforms**

Keywords*— *Social medias, big data, veracity analysis, rumor detection

I. INTRODUCTION

In the era of big data, the growth of information has been incomparable, particularly with the explosive expansion of social media platforms. As in 2024, 4.95 billion people use social media which is equivalent to 61% of the world's population [1]. However, within this abundance lies a critical challenge— "the fourth V" of big data [2], the veracity of data.

A. Rumor detection

One of the primary challenges in the context of data veracity analysis on social medias resides in the fact that information shared on those platforms might reveal misleading elements categorized, according to [8], into three types: fake news, rumors, and other misinformation. Fake news refers to intentionally false shared articles that can be verified, while rumors defined as unverified information often spread on social platforms, varying in their veracity and impact. In addition, we have other misinformation types like clickbait, social spammers, and fake reviews contribute to the complexity of discerning truth from falsehood in online content, rendering it unreliable. Recognizing these challenges underscores the importance to put as a priority the rumor detection practices, in the aim to enhance the quality of data. Before going through specifics approaches applied in order to solve the rumor spread problem, let's commence with a foundational understanding of rumor (ruːmə(r)) defined in [3] as a piece of information, or a story, that people talk about, but that may not be true. It is important to separate rumors from fake news in order to work on rumor detection. Various definitions exist for rumor detection. In certain studies, it is characterized as the process of discerning whether a narrative or online post qualifies as a rumor or non-rumor. When the veracity value of a rumor is determined to be false, certain studies refer to it as a "false rumor" or "fake news". However, several prior investigations provide a more stringent definition for "fake news, «classifying it as a news article intentionally and verifiably disseminating false information.

According to [4], the significant attributes of rumors within the area of social networks serve as compelling reasons for engaging in research in this domain. Firstly, the rapid dissemination of rumors across the network poses a challenge in distinguishing between factual information and misinformation. Secondly, the potential impact of rumors on various aspects of society, leading to social unrest and stress, underscores the need for effective rumor management strategies. Lastly, the ability to trace identified rumors back to their source holds the promise of eradicating potential rumor-generating nodes from the network, offering a valuable avenue for addressing the root causes of misinformation and enhancing the overall reliability of information circulating in social spaces.

B. Approaches categorization

Numerous studies have been conducted on this topic, each one of these studies adopting a distinct approach, most of them mainly focus on one or many of the following:

TABLE I. APPROACHES CATEGORIES

	Rumor detection approaches		
Information used per approach	**Source-based**	**Content-based**	**Propagation dynamics**
	User's information	Textual &visual content	Propagation & network features

1) Source-based: User credibility considered one of the main elements to inspect. It is determined by factors such as registration age, number of followers, and the history of authored posts. Those factors contribute mainly to the comprehensive evaluation of the information source. By considering both community-based reputation and individual user information, these recommendation-based approaches offer a multifaceted assessment of source reliability in the context of merit, achievement, and trustworthiness's

2) Content-based: Content-based approaches, often explained as fact-checking, trying to iteratively calculate and refine the trustworthiness score of a source. Simultaneously, these approaches adjust the belief score of each claimed data in relation to the trustworthiness of the sources endorsing it. Within these methodologies, source quality is initialized and updated based on content belief. It's essential to underscore that this context encircles both textual content and multimedia (visual content). Additionally, several probabilistic models have been proposed to encompass various facets extending beyond source trustworthiness and data belief.

3) Propagation & network dynamics: Propagation-based and also in some studies defined as network-based approaches are central for understanding the dynamics of rumor spread on

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

social media. Rumor circulates through shares and re-shares, forming a diffusion cascade or tree structure. These approaches fall into cascade-based and network-based categories. In the cascade-based methods, cascades are utilized to detect rumors by calculating similarities or generating representations. These models, exemplified by various studies, outperform the feature-based algorithms, although the early detection challenges persist. Network-based methods construct flexible networks to capture rumor propagation indirectly. These networks can be homogeneous, heterogeneous, or hierarchical, with examples leveraging algorithms like PageRank used by Google Search and incorporating network features to enhance detection accuracy

II. RELATED WORK

A. Datasets

The research studies on rumor detection leverage a variety of datasets, each one offers a unique perspectives and challenges. Among the available public datasets, the PHEME dataset stands out as the most used dataset for its collection of rumors and non-rumors tweets from breaking news events, such as the Ottawa shooting, Charlie Hebdo shooting, Sydney siege, Ferguson unrest, and Germanwings plane crash. This dataset, compiled by Zubiaga et al. (2016), contains 5,802 annotated tweets, including 1,972 rumors and 3,830 non-rumors, categorized into subfolders based on event type. Also, from Twitter we have Twitter15 and Twitter16 datasets, hailing from studies by Liu et al. (2015) and Ma et al. (2016, 2017b) respectively, delve into 1,490 and 818 rumors.

Kaggle as a data provider, offer access to datasets from Snopes.com, Emergent.info, and Politifact.com. Those websites provide a structured format for multi-label classification tasks with varying class labels denoting the veracity of rumors.

The Newly Emerged Rumors dataset, captures short-lived rumors, allowing researchers to explore rumor classification and tracking tasks. In addition, we have the Liberia-Ebola 2015 dataset that focuses on rumors during the Ebola crisis, collected through mobile phones by Community Health Workers and journalists. Furthermore, the Credibility Corpus compiles datasets from social media platforms like Twitter and web documents, featuring texts in French and English related to rumors and disinformation. This corpus includes specific rumor-related corpora and a common "rumor indicator" column used for labeling rumors.

Also, in similar research studies, datasets like LIAR, PHEME, Twitter, and Weibo datasets are broadly utilized. The LIAR dataset, sourced from the provider politifact.com API, offers a balanced collection of human-labeled statements categorized into six truthfulness groups. Researchers experiment with the PHEME dataset for rumor detection, focusing on event-centered tweets and conversations initiated by fake news. Some studies opt to collect their datasets, ensuring balance and quality, while others rely on news articles datasets, public resources, or rumor tracking websites like Snopes.com and emergent.info for diverse sources of rumor data.

TABLE II. Datasets Used in Studies for Rumor Detection

Dataset	Information	Format / Size / Source
PHEME	Text, user and propagation features	JSON,6425 rumors, Twitter
WEIBO	Text and user features	JSON, 4646 rumors, Weibo
KAGGLE SNOPES	Text features	CSV, 16.9K rumors, Facebook & Twitter
KAGGLE POLITIFACT	Text, user and propagation	CSV, 2923 rumors, Twitter
NEWLY EMERGED DATASET	Text, user and propagation	XLSX, 106 K rumors, Twitter

B. Methodologies review

In [6] Sushila Shelke & Vahida Attar proposed a combined approach of the content, user and lexical features focusing mainly on the Twitter microblogging. Authors consider the topic as a classification problem focusing on a deep-learning based approach. They categorize the techniques into the recurrent neural network (RNN), convolutional neural network (CNN) and the hybrid models (combining RNN and CNN), articulate the formal presentation of rumor detection in social networks as an event-wise sequence of posts, which is provided as input to the proposed model. In rumor detection, CNN's can analyze textual data by treating it as a spatial pattern, extracting features like word sequences or n-grams. On the other side, we have the RNN's are designed to model sequential data and have memory capabilities, making them suitable for tasks involving temporal dependencies, such as language translation or sentiment analysis. In rumor detection, RNN's can analyze text in a sequential manner, capturing context and dependencies between words to identify potential rumors. The data used for the research was collected then labelled as rumor or non-rumor based on the verification of PolitiFact and Snopes, then preprocessed by applying the most common techniques of data cleaning starting from removing URL's, hashtags, mentions …etc.in order to pull out the lexical features and word embedding. The proposed model presented in the study by the authors, integrates three key components: Word Embedding, Bidirectional LSTM (BiLSTM), and Multilayer Perceptron (MLP). Word Embedding employs a Keras embedding layer in order to convert the tweets into dense vectors, this facilitates the representation of the semantic relationships between words. The BiLSTM model, which is a variant of Recurrent Neural Networks, processes these vectors in both forward and backward directions, capturing contextual information. Meanwhile, the MLP, function as a deep neural network, extracts post-wise features from user, lexical, and content-based aspects. The proposed models include BiLSTM_Embed, where word embedding vectors undergo processing through a BiLSTM model with two dense layers. Lex_PCA utilizes Principal Component Analysis (PCA) to reduce 190 lexical features to 125, implemented with three dense layers. UCL_PCA combines user and content-based features with lexical components, normalizing 145 features through standard scaling. The comprehensive BiLSTM_UCL model merges features from both BiLSTM_Embed and UCL_PCA, resulting in an advanced deep learning architecture for enhanced rumor detection capabilities

The article by Rashed, Kazi, Titya, and Noor [7] provides a similar focus on the user-based and content-based features with a particularity of the user information protection layer. The proposed approach by Rashed, Kazi, Titya, and Noor for rumor detection use a comprehensive set of algorithms and

techniques. The model utilizes the Twitter dataset from the "PHEME Dataset of Rumors and Non-rumors" and employs APIs for data access, with a primary focus on Twitter due to its REST and Streaming APIs. To address the dataset's imbalance, the authors manage to implement a feature-rich selection and extraction process, that categorize features into content-based and user-based. Content-based features include word embedding, subjectivity, sentiment score, and hasMedia, while user-based features are limited to five. Text preprocessing involves lemmatization and the removal of URLs, user mentions, hashtags, punctuation, and white spaces. The core of the proposed model lies in the Rumor detection phase, where various machine learning algorithms are evaluated. This includes popular choices such as Artificial Neural Network (ANN), k-Nearest Neighbor, Support Vector Machine (SVM), Logistic Regression, and Random Forest. The model's performance is extensively assessed using metrics like classification accuracy, Precision, Recall, F-Score, Mean AUC Score with 10-fold Cross-Validation, MCC Score, Confusion Matrix, and ROC Curve. Notably, the authors acknowledge the dataset's small size but strategically leverage it as testing data, safeguarding a large training dataset from unauthorized access. The framework extends beyond the algorithmic choices, incorporating a two-plane architecture (User Plane and Developer Plane) with further division into Module-1 and Module-2 within the Developer Plane. Module1 focuses on securing training data using RSA cryptography, ensuring portability for sharing with authorized third parties. The selection of RSA is justified by its public-key nature, resistance to known attacks, and implementation ease in Python. The proposed framework, not only demonstrates a sophisticated integration of algorithms but also emphasizes the importance of data security and confidentiality in rumor detection efforts.

Similar to the [9] methodology, in [10], Zhenhuang, Hanbing and Yefu focused on the text, and user characteristics mixed approach. They observed a high frequence of rumors, particularly regarding political and social issues, on platforms like Sina Weibo, specially that impact social stability. To address this, they proposed a sentiment analysis approach to assess the emotional polarity of microblog comments, using the proportion of negative emotional comments as a detection feature. Their method combines emotional lexicons, semantic rules, and machine learning for comprehensive feature extraction and sentiment analysis. Additionally, they extracted features from user data, such as authoritative user participation in rumor refutation and user reputation value, as key indicators for detecting rumors.

In the context of the propagation-based methodology, the authors in [11] propose a novel approach for rumor detection that considers both temporal and topological characteristics considering it as crucial element in understanding rumor propagation. They introduce a method for constructing a propagation graph based on non-sequential post propagation on social networks. This is followed then by the development of a representation learning algorithm, PGNN (Propagation Graph Neural Network), which uses gated graph neural networks to learn powerful node representations within the propagation graph. They then present two models for rumor detection: GLO-PGNN, which identifies rumors using global graph embeddings, and ENSPGNN, which calculates prediction probabilities for each node and aggregates them for the final result. Their empirical evaluation on Twitter data shows that these models outperform existing rumor detection methods, especially in early detection scenarios. Their contributions include the explicit construction of propagation graphs, the development of PGNN, and the superior performance of their models in rumor detection tasks and early detection scenarios.

Lin, Xueming, and Caiyan [12] have introduced an approach that addresses the urgent requirement for automated and effective rumor detection methods. This need arises from the swift dissemination and harmful consequences of rumors enabled by Internet technology. It introduces DAN-Tree and DAN-Tree++ models to overcome shortcomings in existing rumor detection methods. DAN-Tree employs a dual-attention network on propagation tree structures, utilizing Transformer encoding blocks to model relationships among posts and focusing on key post nodes and paths. DAN-Tree++ further incorporates user features, considering user's credibility characterized by their profiles. It uses path oversampling to enrich feedback, post-level attention to capture post importance, and path-level attention to identify critical paths in rumor propagation trees. The models are evaluated on various datasets and show superior performance, particularly DAN-Tree++, which achieves the best results. The DANTree++ model consists of three main modules: user feature extraction and encoding, global user feature encoding, and rumor classification, integrating propagation structure and user features for improved rumor detection.

In their study on rumor detection, also on Twitter, the authors in [13] introduce an innovative approach by employing k-nearest neighbor (KNN) and naive Bayes classifier algorithms. The KNN algorithm leverages user-based features, such as the user's account creation date, number of followers, tweets posted, and total favorite counts. It calculates the Euclidean distance between records based on these attributes. While for the naive Bayes classifier, the focus is completely on content-based features, particularly the frequency of words within tweet messages. The methodology includes a thorough pre-processing algorithm that removes URLs, mentions, and hashtags from tweets while retaining and counting rumor keywords. By combining these methods, the study aims to improve rumor detection by using both user-specific and content-specific information, offering a comprehensive data analysis.

In [14] Harun and Bilal propose a comprehensive approach to rumor detection that involves several steps. Initially, the data undergoes preprocessing, including tokenization, removal of stop words, and stemming. Feature extraction follows, with term frequency computation and the creation of a document term matrix. The study applies multiple machine learning algorithms for classification, including One Rule (OneR), Naive Bayes, ZeroR, JRip, Random Forest, Sequential Minimal Optimization (SMO), and Hoeffding Tree. These algorithms are compared based on evaluation metrics such as precision, recall, F-measure, accuracy, ROC area, and PRC area. The study systematically evaluates the performance of each algorithm using different experimental setups, including

the entire dataset, random allocation of training and test data, and cross-validation tests, demonstrating the efficacy of the proposed models in detecting rumors in online social media.

Finally, the article [16] by the authors provides a similar focus on the feature change extraction framework (FCEF) approach to detect rumors. The methodology involves several steps: first, the authors serialize the comment text of each rumor event to obtain values of different dynamic features across sequences. They then use sliding windows of varying sizes to extract local change information, inspired by convolutional neural networks. Simultaneously, they extract global features such as rumor distribution characteristics and outbreak points. The combination of these features, both basic and dynamic, is utilized to train a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel. The parameters of the SVM, specifically C and Gamma, are optimized through Grid Search to enhance the classifier's performance.

III. MODELS PERFORMANCES

An analysis of the effectiveness of the models used becomes essential to comprehend earlier research in the context of our problematic focusing on rumor detection. This chapter provides an extensive and detailed analysis and comparison of the various models used on each research, based on diverse approaches used in previous studies.

The accuracy and the performance of the models in the research [6] are thoroughly evaluated and compared across different scenarios. According to authors, the BiLSTM_UCL model stands out with an impressive accuracy of 97% on both benchmarked and real-world extended datasets. This model combines mainly essential features from user, content-based, and lexical categories, contributing to its high precision and recall values. Comparisons with other models, such as Lex_PCA and UCL_PCA, highlight the remarkable superiority of the BiLSTM_UCL approach, especially in terms of precision, recall, and F1 score for both non-rumor (NR) and rumor (R) classes. Additionally, the research emphasizes the scalability of deep learning models on large datasets, affirming the suitability of the proposed methodology for real-world applications where dataset sizes can vary significantly. Overall, these detailed analyses underscore the effectiveness and potential of the BiLSTM_UCL model in enhancing accuracy and performance in rumor detection tasks.

The model in [4] achieved varying levels of accuracy across different classifiers: K-nearest neighbor (k-NN) at 64.5%, Support Vector Machine (SVM) at 53.5%, Random Forest (RF) at 71.0%, and Naive Bayes at 65.8%. Further analysis using the Random Forest classifier showed that excluding proposed features resulted in a precision of 68.4%, recall of 69.9%, and an F1-score of 68.5%. Including all proposed features improved these metrics to a precision of 69.1%, recall of 70.6%, and an F1-score of 69.8%. Interestingly, excluding the reliability score, although ranked last in importance, led to a slight decrease in precision (68.7%) but an increase in recall (70.2%) while maintaining the F1-score at 68.5%. These comparisons highlight the impact of feature selection on model performance, with Random Forest proving to be the most effective classifier in this context.

According to Rashed, Kazi, Titya, and Noor in [9], model's performance stands out with a focus on content-based features and word embeddings, using only a limited set of user-based features. It employs Random Forest and Artificial Neural Network (ANN) classifiers, achieving impresive accuracy levels of 94% and 91%, respectively. Even the commonly suggested classifiers like SVM, Logistic Regression, and k-Nearest Neighbors performed exceptionally well, surpassing 80% accuracy. Precision, recall, and F1-Score metrics showcase Random Forest's superiority across all measures, reinforcing its effectiveness. The Matthews Correlation Coefficient (MCC) scores, ROC curve, confusion matrix, and Mean AUC Score validate the model's robustness and reliable prediction quality. Additionally, the model's portability and security features, demonstrated through the generation of rumor detection reports and CSV outputs, add practical value. Despite an execution time of around 29.21 minutes, optimizations could further enhance efficiency.

The experiment's in [10] results reveal notable advancements in both emotional classification and rumor detection through various feature sets. When compared to lexicon-based and SVM-based methods, our approach significantly outperformed in judging microblog comment polarity. Moving to rumor detection, experiments with different feature combinations yielded insightful results. Incorporating OLD + PNEC achieved an accuracy of 85.28% for rumors and 84.65% for normal microblogs, while OLD + AUPI and OLD + URV attained accuracies of 81.77%/80.19% and 83.41%/81.52%, respectively. However, the most striking performance improvement was seen with OLD + NEWS, showcasing an accuracy of 86.18% for rumors and 84.62% for normal microblogs, underscoring the effectiveness of these new features.

The performance metrics of the machine learning models in the content-based study [14] are evaluated using precision, recall, F-measure, accuracy, ROC area, and PRC area. The JRip algorithm achieved a precision of 0.990, recall of 0.980, F-measure of 0.985, accuracy of 97.905, ROC area of 0.992, and PRC area of 0.993. ZeroR yielded a precision of 0.686, recall of 0.728, F-measure of 0.814, accuracy of 68.590, ROC area of 0.500, and PRC area of 0.686. Naive Bayes demonstrated a precision of 0.988, recall of 0.988, F-measure of 0.988, accuracy of 98.793, ROC area of 0.999, and PRC area of 0.999. Random Forest reported a precision of 0.991, recall of 0.993, F-measure of 0.992, accuracy of 98.896, ROC area of 0.999, and PRC area of 0.999. SMO obtained a precision of 0.991, recall of 0.991, Fmeasure of 0.991, accuracy of 98.740, ROC area of 0.985, and PRC area of 0.988. OneR produced a precision of 0.803, recall of 0.724, F-measure of 0.639, accuracy of 72.224, ROC area of 0.561, and PRC area of 0.688. Finally, Hoeffding Tree achieved a precision of 0.992, recall of 0.980, F-measure of 0.986, accuracy of 98.110, ROC area of 0.999, and PRC area of 0.999 .

Finally, for the propagation-based models, [12] analyzing the rumor classification performance across Twitter15, Twitter16,

PHEME, and Weibo datasets, the model showed significant improvements. For example, on the Twitter15 dataset, the DAN-Tree model achieved an accuracy of 90.2%, surpassing previous best results by 1.6%, while DAN-Tree++ further improved to 90.9%. On the PHEME dataset, the F1 score increased from 77.2% to 83.0% with the model. Notably, on the Weibo dataset, accuracy improved from 94.3% to 95.8% with DAN-Tree, and DAN-Tree++ achieved 91.3% accuracy on Twitter16. These enhancements highlight the effectiveness of the dual-attention model DAN-Tree and DAN-Tree++ in rumor detection tasks, particularly in incorporating user features for improved performance.

While in [11], The accuracy performance of the proposed model in rumor detection showcases its superiority over traditional methods and state-of-the-art neural network baselines. Traditional methods like SVM-BOW, RFC, and SVM-TS achieved accuracies of 54.3%, 55%, and 45.2%, respectively, indicating their limited effectiveness due to reliance on handcrafted features. In contrast, neural network baselines such as GRURNN (69.5%), BU-RvNN (70.2%), and TD-RvNN (72.7%) demonstrated significant improvements, emphasizing the advantage of neural networks in learning discriminative features. However, the proposed methods, including GLOPGNN (74.3%) and ENS-PGNN (73.1%), surpassed both traditional and neural network approaches, achieving the highest F1 scores.

According to [18] the primary metrics employed include accuracy, precision, recall, and F1-score.The results are presented for different feature combinations: basic features, basic and sentiment features, sentiment and context features, and the full feature change extraction framework (FCEF). For the basic features, the model achieved an accuracy of 0.755, with a precision of 0.82, recall of 0.72, and an F1-score of 0.76. When combining basic and sentiment features, the performance improved significantly, reaching an accuracy of 0.905, precision of 0.90, recall of 0.90, and an F1-score of 0.90. The inclusion of context features along with sentiment features yielded an accuracy of 0.891, precision of 0.90, recall of 0.88, and an F1-score of 0.89. The full FCEF approach demonstrated the highest performance, with an accuracy of 0.932, precision of 0.93, recall of 0.93, and an F1-score of 0.93

TABLE III. Models / Classifiers performance

Article Reference	Publication year	Model/Classifier Name	Accuracy (%)
[6]	2022	BiLSTM_UCL	97.0
[4]	2020	K-nearest neighbor (K-NN)	64.5
		Support Vector Machine (SVM)	53.5
		Random Forest (RF)	71.0
		Naive Bayes	65.8
[7]	2020	Random Forest	94.0
		Artificial Neural Network (ANN)	91.0
		SVM	>80.0
		Logistic Regression	>80.0
		K-nearest neighbor (K-NN)	>80.0
[10]	2018	OLD + PNEC	85.28 (R), 84.65 (NR)
		OLD + AUPI	81.77 (R), 80.19 (NR)
		OLD + URV	83.41 (R), 81.52 (NR)
		OLD + NEWS	86.18 (R), 84.62 (NR)
[14]	2019	JRip	97.905
		ZeroR	68.590
		Naive Bayes	98.793
		Random Forest	98.896
		SMO	98.740
		OneR	72.224
		Hoeffding Tree	98.110
[12]	2023	DAN-Tree (Twitter15)	90.2
		DAN-Tree++ (Twitter15)	90.9
		DAN-Tree (Weibo)	94.3
		DAN-Tree++ (Weibo)	95.8
		DAN-Tree++ (Twitter16)	91.3
[11]	2020	SVM-BOW	54.3
		RFC	55.0
		SVM-TS	45.2
		GRU-RNN	69.5
		BU-RvNN	70.2
		TD-RvNN	72.7
		GLO-PGNN	74.3
		ENS-PGNN	73.1
[16]	2020	Basic Features	75.5
		Basic + Sentiment Features	90.5
		Sentiment + Context Features	89.1
		Full FCEF	93.2

IV. CONCLUSION

To sum up, our investigation into rumor detection techniques has revealed a dynamic field influenced by many strategies, extensive datasets, and exacting performance assessments. Our journey highlights the ongoing evolution and significance of rumor detection in the current digital era, starting with the introduction that emphasizes the urgency of countering misinformation and continuing with the classification of methodologies and algorithms and the comprehensive review of performance metrics like accuracy. This comprehensive strategy ensures strong methods for differentiating fact from lie in the field of information distribution, laying a strong foundation for future developments.

REFERENCES

[1] R. Shewale, “Social media user statistics 2024”. [Online]. Available: https://www.demandsage.com/social-media-users/

[2] L. Berti-Équille and J. Borge-Holthoefer, “Veracity of Data from Truth Discovery Computation Algorithms to Models of misinformation Dynamics”, Qatar Computing Research Institute Hamad Bin Khalifa University pp. 15–20, Jun. 2015

[3] "Rumor," Oxford Learner's Dictionaries. [Online]. Available : https://www.oxfordlearnersdictionaries.com/definition/english/rumour_1?q=rumor.

[4] N. S. Jogalekar, V. Attar, and G. K. Palshikar, “Rumor Detection on Social Networks: A Sociological Approach,” in *2020 IEEE International Conference on Big Data (Big Data)*, 2020

[5] Q. Li, Q. Zhang, L. Si, and Y. Liu, “Rumor Detection on Social Media: Datasets, Methods and Opportunities,” in *Association for Computational Linguistics*, 2019.

[6] S. Shelke and V. Attar, "Rumor detection in social network based on user, content and lexical features," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17347-17368, Mar. 2022.

[7] Md. Rashed Ibn Nawab, Kazi Md. Shahiduzzaman, Titya Eng, and Md Noor Jamal, “Rumor Detection in Social Media with User Information Protection”, EJECE, European Journal of Electrical Engineering and Computer Science Vol. 4, No. 4, July 2020

[8] A. Bondielli and F. Marcelloni, "A Survey on Fake News and Rumor Detection Techniques," *Information Sciences*, 2019, doi: https://doi.org/10.1016/j.ins.2019.05.035.

[9] S. Shelke and V. Attar, "Source detection of rumor in social network – A review," *Online Social Networks and Media*, vol. 9, pp. 30-42, 2019, doi: 10.1016/j.osnem.2018.12.001

[10] Z. Yong, H. Yao, and Y. Wu, "Rumors Detection in Sina Weibo Based on Text and User Characteristics," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2018, pp. 1380-1386, doi: 10.1109/IMCEC.2018.8469406.

[11] Z. W., D. Pi, J. Chen, M. Xie, and J. Cao, “Rumor detection based on propagation graph neural network with attention mechanism,” Expert Systems with Applications, vol. 161, pp. 113595, Jul. 2020, doi: 10.1016/j.eswa.2020.113595. [Online]. Available: https://doi.org/10.1016/j.eswa.2020.113595

[12] L. Bai, X. Han, and C. Jia, “A rumor detection model incorporating propagation path contextual semantics and user information,” Neural Processing Letters, vol. 55, no. 1, pp. 9831–9850, Jul. 2023, doi: 10.1007/s11063-023-11229-w. [Online]. Available: https://doi.org/10.1007/s11063-023-11229-w

[13] R. Dayani, N. Chhabra, T. Kadian, and R. Kaushal, “Rumor detection in Twitter: An analysis in retrospect,” in Proc. IEEE ANTS, 2015, doi: 10.1109/ANTS.2015.1570205805.

[14] H. Bingol and B. Alatas, "Rumor Detection in Social Media Using Machine Learning Methods," Journal of Information Science and Engineering, vol. 36, no. 2, pp. 287-301, 2020.

[15] Y. Liu and R. Yang, "Rumor Detection of Sina Weibo Based on MCF Algorithm," in International Conference on Computing and Data Science (CDS), 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9275940.

[16] Z. Meng, S. Yu, R. Li, G. Jiang, and Y. Song, "Dynamic Features Based Rumor Detection Method," 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9164761.

turnitin Page 47 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

The Prospective Threat Vector of a Bounded and Controllable Optimized Computational Approach for Spatio-Temporal Knowledge Graph Completion

S. Chan
VTIRL, VT/I-PAC
Orlando, USA
stevec@i-pac.tech

***Abstract*—There is a presumption by some that the removal of certain materials from online repositories can increase "security through obscurity." This potentially specious reasoning (as snapshots of these materials may still be available via various digital preservation archives on the surface web as well as the dark web) tends to also be coupled with an underestimation regarding the State-of-the-Art (SOTA) for Spatio-Temporal Knowledge Graph Completion (STKGC) (i.e., the positing of missing nodes, links, etc.) and the possibility of effectively computing the minimum/optimal Control Energy Cost (CEC) for controlling a Large-Scale Complex Network (LSCN). Prospective vulnerable Real World Scenario (RWS) LSCN include power grids and Artificial Intelligence (AI) compute clusters. This paper discusses a prototype Key Node Identification/Assessment (KNIA) module, which successfully undertook STKGC for a power grid and computed the optimal number of Control Signals (CS_{opt}) operating on an optimal number of Key Control Driver Nodes ($KCDN_{opt}$) at an optimal CEC (CEC_{opt}) over an optimal Elongated Temporal Span (ETS_{opt}) so as to constitute prospective meaningful control over the involved digital rendition LSCN. A bespoke architectural construct, in the form of a Hypergraph-Induced Infimal Convolutional Manifold Neural Network (H-IICMNN), was utilized to resolve the aforementioned KNIA-related CS-KCDN-CEC-ETS optimality problems. By discerning these key pathways, defensive bulwarks for mitigation against certain prospective threat vectors can be formulated and instantiated.**

Keywords—algorithms and systems for big data analytics, knowledge extraction and discovery, machine learning for big data, real-world and large-scale practices of big data.

I. INTRODUCTION

Knowledge Engineering (KE) is an often-used rubric to encompass the range of KE constituent elements, such as from acquisition through justification. As Lu notes, conventional KE "and knowledge-based software engineering have undergone fundamental changes where the network plays an increasingly important role" [1]. Along this vein, Arruda asserts that "complex networks have been found to provide a good representation of the structure of knowledge, as understood in terms of discoverable concepts and their relationships" [2], and Schwartz underscores the current trend of "data mining and knowledge discovery using complex networks" [3]. After all, from the COVID experience, the significance of being able to locate the epicenter of the spread of an infectious disease is understood; similarly, with cyber in the news, the ability to identify vulnerable nodes that can potentially yield control of a mission-critical network (e.g., power grid) or lead to the manipulation of model training (e.g., AI compute cluster) has profound societal implications [1].

Taking the power grid as just an introductory example, the involved Large-Scale Complex Network (LSCN) is often subject to a variety of temporally-shifting conditions. For example, in addition to weather events and technical events, there are also a variety of anthropogenic-related events (e.g., high-impact events within conflict regions). In these cases, the complex network changes are likely to be dynamic and ongoing, so there needs to be careful consideration given to the type of Knowledge Graph (KG) leveraged. Static KGs (SKGs) do not always suffice given the lack of temporal considerations. Temporal KGs (TKGs) are not necessarily fit for purpose either, as they disregard spatial considerations. Spatio-Temporal KGs (STKGs) are a closer approximation to the desired expressiveness, and Interdonato uses the term "feature-rich network" to describe the envisioned "expressive power" beyond that typical "complex network topologies" [4].

A. Incompleteness of SKG, TKG, and STKG

Regardless of the type of KG, they are typically incomplete — devoid of a number of invaluable Triples, Quadruples, and/or Quintuples (TQQ) — which has a non-trivial impact on the involved model(s) and ensuing performance. The goal, then, is to undertake KG Completion (KGC), such as STKG Completion (STKGC), so as to determine the "unobserved" TQQ and facilitate "sufficient inference," as noted by Zeb [5]. Robust KGC can, potentially, facilitate reducing the inference load and the accompanying AI Energy Consumption (AEC). This is important because, as Luccioni notes, "inference happens far more frequently than model training" and higher-order tasks (e.g., advanced analysis, such as anomaly detection) tend to have a much higher AEC involved [6]. By way of example, Schneider notes in its "Overview of AI workloads in data center," that for 2023, the AEC was at approximately "20% Training, 80% Inference," and it estimates that by 2028, it will be at about "15% Training, 85% Inference" [7]. Cloudflare notes that "the better trained a model is, and the more fine-tuned it is, the better its inferences will be," which should lead to a reduction in the number of inference passes and a lower AEC [8]. Hence, a robust KGC can likely have a profound impact on AI-based computation by serving as an inference AEC optimizer.

B. STKGC For Real World Scenarios (RWS)

To underscore the importance of apropos and robust KGC, such as STKGC, a RWS is reviewed. A well-known think tank, Atlantic Council, had noted that an aid agency had "reportedly sent a large shipment of equipment incompatible with" … the aid recipients' power grid, and the equipment was being stored until "the logistics of returning it are worked out" [9]. Taking the cited case, if appropriate Key Node Identification/Assessment (KNIA) and the associated technical granularities are not able to be determined for even SKG paradigms, then TKG paradigms would be too far afield (and STKG paradigms would be even farther afield). A prototype KNIA module leveraging STKGC was developed for experimentation/Proof of Concept (POC) purposes.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

C. Prospective Efficacy of STKGC

To demonstrate the potential efficacy of STKGC, we provide a rudimentary exemplar; initially, the base dataset considered was the European Network of Transmission System Operators for Electricity (ENTSO-E), which strives to delineate the power grid mapping for the various European countries; however, at least for certain grids depicted by ENTSO-E, the depicted nodes (for those certain grids) are anonymized (i.e., there are neither substation nor generation node names). Moreover, the ENTSO-E power grid map, at least for select countries, does not include a number of the future substation nodes and future generation nodes, which are needed for a more meaningful analysis. For the select countries of interest for the POC, segments of the 110 kV power lines connected to the generation sites are also missing. Hence, in isolation, the ENTSO-E power grid map did not suffice (nor was it necessarily intended to do so) for the STKGC experimentation and POC vulnerability analysis undertaken, which leveraged hybridized spatio-temporal considerations. For example, if the depicted area is in a conflict region, conditions may fluctuate temporally, and TKGC might be critical to employ. Along this vein, enrichment of the considered base dataset could come in many forms ranging from the use of the Sentient Hyper-Optimised Data Access Network (SHODAN) search engine for insights into the Internet of Everything (IoE) (e.g., for context, such as endpoint insights) to snapshots via satellite images and other mappings residing within repositories, such as the NASA Earth Observing System (EOS) (e.g., Lights at Night, Landsat Image Gallery, etc.), as well as various archives (for baselining and/or reference point purposes). Although certain materials (e.g., deemed "sensitive") may have been deliberately removed, select snapshots are still available via various digital preservation paradigms (e.g., Internet Archive's Wayback Machine). Furthermore, the use of certain Computer Vision (CV) tools/methods, Natural Language Processing (NLP) approaches, and Generative KG Construction (GKGC) methods were invaluable in this regard.

This paper is structured as follows. Section I introduced the potential capacity for STKGC to ascertain the "unobserved" TQQ as well as to serve as a computational accelerant; the section also provides some context for a power grid-related case study, whose focus is to demonstrate the possibilities of an optimized computational approach for STKGC. Section II presents relevant background information regarding the intricacies of the LSCN controllability challenge and reviews the seeming latent stability effects of "Higher-Order Networks" (HON). Section III provides some theoretical foundations for the involved KNIA-related experimentation; it presents a bespoke H-IICMNN for dense/homogeneous spatio-temporal LSCN. Section IV provides some preliminary results, puts forth some envisioned future work, and the acknowledgements close the paper.

II. Background

To best contextualize the prospective threat vector of a high efficacy computational approach for STKGC, several concepts are presented in a logical progression starting with the potential latent stability effect of HON and concluding with the insights offered by complex manifolds.

A. Higher-Order Networks (HON)

For KGC, discriminative methods (a.k.a., conditional methods that discern boundaries among labels, classes, etc.) endeavor to, by way of example, "predict the possible label" (e.g., node name, line segment name, etc.), as asserted by Ye [10]. As noted by Wei, discriminative methods focus on discerning elements of the TQQ so as to "efficiently construct large-scale" KGs, "which often require multiple models to process" [11]. Indeed, for some cases, an ensemble and/or cascading succession of models might be necessary. This is especially instrumental given the contemporary findings regarding "Multi-Layer Networks" (MLN) and HON, as noted by Lu [1]. For example, Grilli had found that HON interactions have a stabilizing influence within LSCN [12], and the existence of HON nicely explains many RWS.

B. Influence Dominating Sets (IDS)

The point of HON is well taken considering the phenomenon of IDS, such as exhibited by the Bak–Tang–Wiesenfeld (BTW) sandpile effect of non-equilibrium systems. IDS are classified as Positive Influence Dominating Sets (PIDS) and Negative Influence Dominating Sets (NIDS), which must be considered in the aggregate for the overall IDS.

C. Minimum Dominating Set Problem (MDSP)

Continuing on this point, overarching constraining functions, effectuated by IDS, such as exemplified by the notions of "keiretsu," "chaebol," and "qiyejituan" constructs (e.g., a set of entities with "dotted-line," interlocking degrees of affiliation), among others, segue to the beginnings of what Nguyen notes as the "MDSP," which "deals with determining the smallest dominating set of a given graph" [13]. However, an approach that "ensures controllability" that is "equivalent to solving a combined maximum matching" (wherein the matching — a repertoire of edges that do not share common nodes — with the greatest number of edges is referred to as a maximum matching) as well as MDSP is a different matter entirely, as noted by Alizadeh [14].

D. Minimum Controllability Problem (MCP)

The resolving of such MDSP (more suited for static LSCN) does not progress to the level of MCP (more suited for dynamic LSCN), as pointed out by Terasaki; whether MDSP or MCP, it is still just one beginning step [15]. Lin added that "the system is structurally controllable if and only if a connection condition ("an *infimal* strongly connected component…which is not an input vertex," also known as a "Critical Connection Component" or CCC) and a rank condition (a requisite and satisfactory condition for the m^{th} equation to be ascertained) are both satisfied" [16][17].

E. Efficient Controllability Problem (ECP)

A more complicated problem, among others, centers upon the ECP, such as noted by Gokler [18], which contends with minimizing the number of requisite CS as well as minimizing the requisite CEC, as underscored by Lindmark [19]. These notions are intertwined, for Chen asserts that "if the number of" CS "is small, the" CEC "demanded…could be prohibitively high;" conversely, the CEC "is reduced exponentially as the number of" input CS increases [20]. Axiomatically, just as an exorbitantly high CEC would be impractical to effectuate, controllability for only a limited temporal span may not be meaningful or suffice for the intended physical controllability Command & Control (C2) paradigm. After all, practical controllability has the criteria of persistence over an ETS so as to be able to effectuate actual/effective control when needed/desired. It would then seem that the optimality problems at hand revolve around an optimal number of CS (CS_{opt}) serving as an IDS on an optimal

number of KCDN ($KCDN_{opt}$) at an optimal CEC (CEC_{opt}) over an ETS_{opt} so as to constitute meaningful control, as noted by Gao [21]. To tackle the described optimality problems, MLN (with its constituent HON) need to be discerned so as to be further considered. This particular discernment leverages: (1) the informative nature of Co-Evolution Networks (CEN), and (2) the insights provided by complex manifolds.

F. *Informative Nature of CEN*

First, delineation of adjacent and/or MLN paradigms of the involved KG, via the vantage point of, for example, CEN (wherein networks evolve together), can be quite informative; As a case in point, Zhang's example is quite intuitive, for he discusses the co-evolution of Electric Vehicles (EVs) predicated upon "the reliability on distribution networks" [22]. Likewise, there has been a co-evolution of a "Smarter Grid" paradigm predicated upon the reliability of available communications networks. Accordingly, KGC can be facilitated by, say, knowledge of either the private and/or Service Level Agreement (SLA) communications network interoperating alongside an involved power grid (i.e., CENs). Likewise, knowledge of the involved networks across various jurisdictional/functional demarcation boundaries (e.g., generation, transmission, distribution) can also be quite insightful for KGC in terms of leveraging further information for the discerning and/or extracting of TQQ.

G. *HON and Complex Manifolds*

Apart from the informative nature of CENs, second, the stabilizing influence of HON has been extensively studied by Battiston and others, and the characterization of HON topologies can be via more complex representations (e.g., hypergraphs) [23] as well as "complex hypergraphs ('hypergraphs of hypergraphs' or chygraphs)," as introduced by Vazquez [24], self-referential hypergraphs ("ubergraphs"), as noted by Joslyn and Nowak [25], and multiplex hypergraphs ("a set of hypergraphs…with the same set of vertices"), as noted by Sun and Biaconi [26]. Ultimately, HON is comprised of "both hypergraphs and Simplicial Complexes" (SC) [26]. Despite the seeming variations, according to Battiston, much of the literature regarding LSCN research centers upon "pairwise interactions" [27]. Yet, RWS typically involve the interplay of "groups of three or more units" [27]. As the referenced HON structures can dramatically influence not only their neighborhood locales, but also the global environs (e.g., the previously discussed BTW effect), an enhanced understanding of these HON structures becomes critical. This segues into the need to suitably characterize HON topologies, such as by way of a "Combination of 'Complex Manifolds'" (CMs), as Ding suggests [28]. Voisin describes a CM as one that has "complex-valued coordinates (called holomorphic coordinates)" assigned to positions on a manifold [29]. CMs can provide invaluable insights, and "a physical system embedded on a twisted topological complex manifold" can bring out "fundamental physical properties of an unknown system," such as "if and when" a "system is undergoing a *phase transition*" [30]. When this is conjoined with the BTW principle described in Section IIB and set against the described LSCN controllability/uncontrollability optimality problems, the impact of existing HON topologies and their CMs on the involved CS-KCDN-CEC-ETS becomes axiomatic [28][29].

III. Theoretical Foundations & Experimentation

A. *Controllability Gramian*

As noted by Klickstein, the minimum CEC (CEC_{min}) "can be characterized by the controllability Gramian" [31]. For the desired "physically controllable case," the involved "Gramian matrix should be well-behaved" (i.e., the condition number — sensitivity to perturbations — and the CEC are not impractically large) [32]. This is in contradistinction to the case for which the Gramian matrix is ill-conditioned (i.e., the condition number and CEC are indeed impractically large), as noted by Wang. For the latter case, the LSCN is not able to arrive at the "final state in the prespecified time within a predefined precision" [32]. As noted by Lindmark, a viable strategy for contending with the Gramian matrix is paramount, as some stategies can only be "computed in closed form ... when the time of the transfer tends to infinity" and physical controllability will not manifest [19]. This accentuates the case for CS augmentation and/or accelerant approach vectors so as to improve the likelihood of physical controllability (as contrasted to theoretical, mathematical controllability). This is particularly important, as more robust and accurate controllability is desired for dense/homogeneous LSCN (vice sparse/heterogeneous LSCN) with sub-LSCN, as noted by Zhou [33]. As discussed in Section IIB, Zhang noted that temporal LSCN, which exhibit link temporality — a comparable paradigm to "attaching a virtual driver node to that link" — tend to be more physically controllable [34]. This is significant, in CS terms, as one set of KCDN can influence another set of KCDN so as to effectuate Phase 0 shaping of not only the involved LSCN, but also peer LSCN and/or HON LSCN to a particular interim state. It then follows that the ultimate desired state is more likely to be attained.

B. *Inverse Gramian*

The degree of success for the CS base candidate set and/or CS augmentation set, which can be collectively construed to be CS_{opt}, is contingent upon the diffusiveness/permeability of the LSCN, as asserted by Ludice [35]; in turn, this constitutes a potential indicator of the susceptibility for LSCN controllability. In this regard, when the LSCN is in an uncontrollable state, the inverse Gramian is not present. In contrast, when the LSCN is in a controllable state, the inverse Gramian does indeed exist. Along this vein, a corresponding Vanishing-Moment Recovery (VMR) matrix (e.g., tight wavelet frames) is a suitable approximation of the inverse Gramian and "guarantees *n* vanishing moments of the irregular framelets" [36]. On this point, a goal, among others, of the experimental testbed, which will be described in Section IIID, is to well handle these "wavelet tight frames with *n* vanishing moments," as noted by Viscardi [36]. Abebe observes, "as the number of vanishing moments increases, the polynomial degree of the wavelet also increases" [37]; in turn, as the degree increases, the involved underlying graph becomes smoother. The potential advantage of this is that, according to Grochenig, "wavelet tight frames can," therefore, "be derived from any multiresolution analysis" [38]; this segues to the discerning of LSCN collective phenomena.

C. *Percolation and the Giant Component of a LSCN*

Accordingly, the notions of *Percolation* and the *Giant Component* of the LSCN should be discussed. Sun notes that *Percolation* "predicts the fraction of nodes in the *Giant Component*" of a LSCN, and "having a non-zero *Giant Component* is the minimal requisite for observing collective

phenomena on networks, emerging from" … diffusiveness/permeability, etc. [26]. Section IIA had noted that the latent stability and IDS characteristics of HON nicely account for many RWS. In addition, Section IIG had noted that hypergraphs and SC were constituent elements of HON. Of significance, Zhang notes the HON interactions "shape collective dynamics differently in hypergraphs and" SC [39]. In fact, Zhang asserts that HON interactions "increase degree heterogeneity in" SC while HON interactions tend to "decrease degree heterogeneity in [random] hypergraphs" [39]. The insights provided by these two constituent elements of HON are not dissimilar to the insights gleaned via the paradigm of CEN. Furthermore, taking SC in isolation, Lee asserts that SC representation "is an elegant framework for representing the effect of complexes or groups with" HON interactions, particularly for RWS [40] and has been "shown to reveal a rich *phase [transition]* diagram" for "link percolation" [26]. This dovetails with the insights proffered by the complex manifold representation of the involved HON topology, and Lee further notes that *Homological Percolation Transition (HPT)* is well suited to reflect HON interactions as well [40]. After all, "homological [e.g., likeness in structure] percolation [e.g., LSCN behavior over time with the addition of various nodes and/or links] has been shown to well characterize the emergence of a non-trivial homology for" HON topologies [26][40]. Sun further articulates the value-added proposition of the SC and hypergraph interplay (in a "CEN"-like fashion) by noting how multiplex hypergraphs tend to reveal a "rich set of phenomena" [26].

D. Experimental Testbed Architectural Modifications

Jin noted that when a prototypical Deep Convolutional Neural Network (DCNN) is confronted with "large intra-class variations, the performance of the traditional [D]CNN models degenerates dramatically" [41]. As previously discussed in Section IIIC and as noted by Gao, complex manifolds can be invaluable for "data preserving" the related HON topologies [42]. As HON "are effectively modeled by the hypergraph" (as well as simplical complexes), Jin posited that a Hypergraph-induced Convolutional Manifold Network (H-CMN) can improve the "representation capacity of the DCNN for the complex data" [41]. Huang also noted that hypergraphs were quite good for "attribute predicting" [43]. Thus, along this hypergraph vein, the utilization of a specialized derivative of the H-CMN, a Hypergraph-Induced *Infimal* Convolutional Manifold Neural Network (H-IICMNN), as opposed to a prototypical general-purpose DCNN, leads to a lower AEC, as Luccioni's experimentation demonstrated that the utilization of "multi-purpose models for discriminative tasks" has a higher AEC (by about 2-3x to 5-7x) when "compared to task-specific models for these same tasks" [6]; as H-IICMNN is more task-specific, it has a lower concomitant AEC. Also, as Commault and Dion as well as others have noted that MCP (and ECP, etc.) are NP-Hard [16], our prior experimentation had heavily relied upon a Graph Convolutional Network (GCN)-Bidirectional Long Short-Term Memory (BiLSTM)-Graph-Attention-Network (GAT) mechanism along with a Robust Convex Relaxation (RCR)-based Deep Convolutional Neural Network (DCNN) Generative Adversarial Network (GAN) (DCGAN)-DCNN-1,2,3,4 amalgam (GCN-BiLSTM-GAT & RCR-DCGAN-DCNN-1,2,3,4) to address the involved NP-Hard problems. With regards to the GCN-BiLSTM-GAT, Zhang affirmed the "expressive power" of GCN [44]. Siami-Namini affirmed the use of the BiLSTM for its "better predictions," such as "in longer prediction horizons" over "regular LSTM-based models" [45][46]. Hamilton affirmed the use of the GAT for its computational efficiency [47]. With regards to the RCR-DCGAN-DCNN-1,2,3,4, the architectural construct was quite adept at handling RCR; the DCNN-1 functioned as the key solver for the involved RCR optimization problems, DCNN-2 functioned as the key solver for the non-convex problems inadvertently spawned by the RCR, DCNN-3 functioned as the key solver for certain modified involved functions, DCNN-4 functioned as a numerical stability stabilizer for the construct, and a DCGAN functioned as a mitigator against mode failure. However, as discussed in Section IIID, a task-specific construct (such as in the form of a bespoke derivative of the H-CMN [41]), the H-IICMNN, could — as Luccioni posits — have a lower AEC than the less task-specific RCR-DCGAN-DCNN-1,2,3,4. Thus, the architectural construct was modified for the current version of the POC, which is still very much a Work-In-Progress (WIP). While the original version strictly employed a succession of winnowing Continuous Wavelet Transform (CWT)-like convolutional filters, which ultimately segued to a CWT PyWavelet schema, the experimentation for this paper leveraged Bauschke's notion of *infimal* convolution for "fundamental convexity-preserving operation[s] for functions" [48]; DCNN-1,2,3 were supplanted with the more task-specific H-IICMNN *infimal* convolution mechanisms to serve as "efficient solvers," as noted by Lambert [49]. By taking this approach, the involved problem is *bounded*, and as a RWS, it belongs to the class of *controllable* network topologies (as contrasted to an uncontrollable class discussed by Aguilar and others [50]). As noted by Wang, there is a great distinction between mathematical and actual/effective controllability for a RWS (and CS augmentation constitutes an additional IDS consideration [51]); along the vein of Zhang's reasoning, prospective CEC reduction, as the link weight fluctuates temporally, can be exploited [34]. A contribution of the bespoke task-specific H-IICMNN was to enhance discernment via a more balanced operationalization since, in the case of RWS, "the Gaussian assumption usually does not hold" [41]. In contrast to the [moderate-tailed] Gaussian distribution, the long-tail distribution tends to be prevalent "in KGs" [52], and "strongly unbalanced data with a long-tail is ubiquitous in numerous domains and problems" while "learning with unbalanced data causes models to favor head classes" [53][54]. Hence, the utilized STKG Embedding (STKGE), to achieve the STKGC, needs to be well balanced across both classes (i.e., head and tail). Certain promising techniques were extended, via He's framework, the Type-augmented Knowledge [Graph] Embedding (TaKE), which "can be combined with any traditional KGE models" "under no explicit type information supervision" and can facilitate "both type constraint and type diversity with low time and space complexity" [55]. A STKG with a Type-Sensitive (TS) extension (a.k.a., TS-STKG) is referred to as a T2S2KG, and T2S2KGE is the generic form, wherein KGE is replaced with the extended model. It was determined that T2S2-DistMult's balancing performance was inferior to that of T2S2-ComplEx (wherein ComplEx is an extrapolation of DistMult [56]). Likewise, it was noted that T2S2-HyTE (an extension of HyTE, which is an extension of TransH) was inferior to that of T2S2-Hybrid-TE (wherein Hybrid-TE is a hybridization of TransD and HyTE) [57]. Hence, T2S2-ComplEx and T2S2-Hybrid-TE were the models utilized for the WIP POC KNIA module's T2S2KGE, and the performance is shown below.

TABLE I. HEAD/TAIL PERFORMANCE FOR T2S2KGE TECHNIQUES

	Predicting Head Entity				Predicting Tail Entity			
T2S2KGE Technique	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
T2S2-ComplEx								
T2S2-Hybrid-TE								

The progression of colors follows the ROYGBIV sequence (i.e., orange denotes worse performance while the darker shades of green denote better performance of the STKGE Technique against the various types of KG relationships (e.g., 1-to-1, 1-to-N, N-to-1, and N-to-N).

IV. REFLECTIONS

The main output of this paper is that of a H-IICMNN approach, as a contribution to the challenge of KNIA-related CS-KCDN-CEC-ETS optimality problems as well as to the challenge of STKGC. Both represent non-trivial issues in the arenas of knowledge extraction and discovery within High Dimensional Big Data for RWS. The enhanced discernment offered by the H-IICMNN approach should not be underestimated as lowering AEC can potentially have profound implications on AI computational approaches. The experimentation herein not only operationalized the WIP POC KNIA module, but it also paved the way for gleaning insights from HON, IDS, MDSP, MCP, and ECP; it leveraged the revealing nature of CEN and CMs (characterizing HON topologies). In particular, the CMs provided enhanced discernment of *phase transitions*; after all, the SC of the HON lends to a rich understanding of *phase transitions*, and the hypergraphs of the HON also exhibit *phase transitions* (in accordance with the hyperedges of size *k*). HPT lends to a repertoire of *transition* insights as well. The well-behaved Gramian matrix, the inverse Gramian, and the VMR all constitute indicators that the LSCN is likely in a controllable state (with suitable diffusiveness/permeability) [58]; this, then, is suitable for the *Percolation* and *Giant Component* prerequisites "for observing collective phenomena on networks, emerging from" suitable diffusiveness, permeability, etc. [26]. Attaining the discernment capability over the discussed observational space segues to more robust STKGC, which can, potentially, facilitate reducing the anticipated inference load and the accompanying AEC. The WIP POC KNIA module leveraged such STKGC for experimentation/POC purposes and exhibits promise.

A. Implications of Robust Discernment Capabilities

The societal implications for efficiently resolving the discussed optimality problems are non-trivial. If the referenced optimality problem(s) can be readily solved, and certain LSCN (and their subordinate LSCN) can be efficiently controlled at a relatively low CEC, that could very well put, by way of example, production clusters engaged in AI/Machine Learning (ML) training at heightened risk. In a number of aspects, this may be as significant as the realm of cryptographic methodological approaches (e.g., Rivest, Shamir, Adleman or RSA, etc.) facing prospective elevated vulnerability from the rapidly advancing state of efficient resolution of factorization and other similar (e.g., discrete logarithmic) SOTA advances in computing (e.g., quantum). The associated financial implications and the impact on the viability of future AI systems can be quite profound. After all, as illuminated by Oligo, an application security firm, the Anyscale Ray (a compute platform that facilitates constructing, fine-tuning, training, and scaling AI/ML applications) 2.6.3 and 2.8.0 vulnerability currently has a Common Vulnerability Scoring System (CVSS) Base Score of "9.8 Critical" (although this is disputed, and it is "awaiting reanalysis") for Common Vulnerabilities and Exposures (CVE)-2023-48022, as listed on the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) portal [59][60]. Against this sobering backdrop, the prospective control and potential ensuing tampering with AI/ML models at the training and/or fine-tuning phases constitutes a very expensive vulnerability. After all, it has been reported that GPT-4 "cost over $100 million" to train, so any manipulation thereof would not be desirable [61]. The tampering with/neutering of AI/ML model training can impact the downstream inferencing cost, such that Advanced Analytics Technologies (AAT) can become cost prohibitive/ineffective in their missions, thereby further exacerbating the defense. In a fashion, the described paradigm is not dissimilar to the scenario of hidden defects/failures, wherein the involved LSCN is compromised at the onset.

B. Discerning Vulnerabilities for Mitigation Enhancement

The aforementioned backdrop served as the impetus for the POC and the related experimentation. The WIP POC KNIA module, which leverages TKGC/STKGC CEN for ECP resolution, was demonstrated during Distributech on 28 February 2024 as well as at a Sandia National Laboratories technically co-sponsored venue. The theoretical foundations were extensively peer-reviewed at these venues. The goal of the POC KNIA module was to discern potential vulnerabilities such that a more robust mitigation paradigm can be effectuated. For example, Chen had observed that CEC could be dramatically reduced when CS augmentation (e.g., KCDN) could be optimally effectuated while minimizing the path lengths (the longest of these path lengths is referred to as the Longest Control Chain or LCC) from KCDN to non-KCDN, via optimal placements of the involved CS [20]. By identifying the involved KCDN, the LCC and CEC can be deliberately increased, so as to enhance the involved mitigation. Future work will involve more qualitative and quantitative experimentation and benchmarking.

ACKNOWLEDGMENT

This paper is part of a series of papers under a Quality Control Program (QCP) implemented by the Quality Assurance/Quality Control (QA/QC) unit — attached to the Underwatch initiative of VTIRL, VT — for I-PAC.

REFERENCES

[1] J. Lu, G. wen, R. Lu, Y. Wwang, and S. Zhang, "Networked Knowledge and Complex Networks: An Engineering View," *IEEE/CAA J. of Automatica Sinica*, vol. 9, pp. 1366-1383, Aug. 2022.

[2] H. Arruda, F. Silva, L. Costa, and D. Amancio, "Knowledge Acquisition: A Complex Networks Approach," *Info. Sci*., vol 421, pp. 154-166, Dec 2017.

[3] G. Schwartz, "Complex networks reveal emergent interdisciplinary knowledge in Wikipedia," *Humanities and Social Sci. Commun*., vol. 8, pp. 1-6, May 2021.

[4] R. Interdonato, M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, and A. Sala, "Feature-rich networks: going beyond complex network topologies," *Appl. Netw. Sci*., vol. 4, pp. 1-13, Jan 2019.

[5] A. Zeb, S. Saif, J. Chen, A. Haq, Z. Gon, and D. Zhang, "Complex graph convolutional network for link prediction in knowledge graphs," *Expert Syst. with Appl*., vol 200, Aug 2022.

[6] A. Luccioni, Y. Jernite, E. Strubell, "Power Hungry Processing: Watts Driving the Cost of AI Deployment?" *Arxiv.org* [Online], Nov 2023. Available: https://arxiv.org/abs/2311.16863.

[7] Victor Avelar et al., "The AI Disruption: Challenges and Guidance for Data Center Design," *Schneider Electric* [Online]. Dec 2023.

Available: https://download.schneider-electric.com/files?p_Doc_Ref=SPD_WP110_EN.

[8] "AI inference versus training: What is AI inference," *Cloudflare* [Online]. Available: https://www.cloudflare.com/learning/ai/inference-vs-training/.

[9] S. Jayanti, "Winter is coming: Is Ukraine's power grid ready for new Russian attacks?" *Atlantic Council* [Online], Aug 2023. Available: https://www.atlanticcouncil.org/blogs/ukrainealert/winter-is-coming-is-ukraines-power-grid-ready-for-new-russian-attacks/.

[10] H. Ye, N. Zhang, H. Chen, and H. Chen, "Generative Knowledge Graph Construction: A Review," *Proc. of the 2022 Conf. on Empirical Methods in Natural Lang. Process*., pp. 1-17, Dec 2022.

[11] Z.Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," *Proc. of the 58th Annu. Meeting of the Assoc. for Comput. Linguistics*, pp. 1476–1488, Jul 2020.

[12] J. Grilli, G. Barabás, M. J. Michalska-Smith, and S. Allesina, "Higher-order interactions stabilize dynamics in competitive network models," *Nature*, vol. 548, pp. 210–213, Jul 2017.

[13] M. Nguyen, M. Ha, D. Nguyen, and T. Tran, "Solving the k-dominating set problem on very large-scale networks," *Comput. Social Netw*., vol. 7, pp. 1-15, 2020.

[14] S. Alizadeh, M. Posfai, and A. Ghasemi, "Input node placement restricting the longest control chain in controllability of complex networks," *Sci Rep*., vol. 13, pp. 1-14, Mar 2023.

[15] S. Terasaki and K. Sato, "Minimal Controllability Problems on Linear Structural Descriptor Systems," *IEEE Trans. on Autom. Control*, vol. 67, no. 5, pp. 2522-2528, May 2022.

[16] C. Commault and J. Dion, "The single-input Minimal Controllability Problem for structured systems," *Syst. & Control Lett*., vol. 80, pp. 505-55, Jun 2015.

[17] C. T. Lin, "Structural controllability*," IEEE Trans. Automat. Control*, vol. 19, no. 3, pp. 201–208, 1974.

[18] C. Gokler, S. Lloyd, P. Shor, and K. Thompson, "Efficiently Controllable Graphs," *Phys. Rev. Letter*, vol. 118, pp. 1-5, Jun 2017.

[19] G. Lindmark and C. Altafini, "Minimum energy control for complex networks," *Sci. Rep.,* vol. 8., pp. 1-14, Feb 2018.

[20] H. Chen and E. Yong, "Optimizing Target Nodes Selection for the Control Energy of Directed Complex Networks." *Sci. Rep*., vol. 10, pp. 1-14, 2020.

[21] L. Gao, G. Zhao, G. Li, L. Deng, and F. Zeng, "Towards the minimum-cost control of target nodes in directed networks with linear dynamics," *J. of the Franklin Inst*., vol 355, pp. 8141-8157, Nov 2018.

[22] D. Zhang, Y. Kang, L. Ji, R. Shi, and L. Jia, "Coevolution and Evaluation of Electric Vehicles and Power Grids Based on Complex Networks," *Sustainability*, vol 14, pp. 1-15, Jun 2022.

[23] F. Battiston, "The physics of higher-order interactions in complex systems," *Nature Phys*., vol. 17, pp. 1093-1098, Oct 2021.

[24] A. Vazquez, "Complex hypergraphs," *Phys. Rev. E*, vol. 107, pp. 1-8, Feb 2023.

[25] C. Joslyn and K. Nowak, "Ubergraphs: A definition of a recursive graph structure," *Arxiv.org* [Online], Apr 2017. Available: https://arxiv.org/abs/1704.05547.

[26] H. Sun and G. Bianconi, "Higher-order percolation processes on multiplex hypergraphs," *Phys. Rev. E.*, vol. 104, pp. 1-17, Sep 2021.

[27] F. Battiston et al., "Networks beyond pairwise interactions: Structure and dynamics," *Phys. Rep*., vol. 874, pp. 1092, 2020.

[28] J. Ding et al., "Artificial Intelligence for Complex Network: Potential, Methodology and Application," *arXiv.org* [online], Feb 2024. Available: https://arxiv.org/abs/2402.16887.

[29] C. Voisin, "Complex Manifolds," *Hodge Theory and Complex Algebraic Geometry I*, pp. 38-62, Jan 2010.

[30] "Reading the physics hiding in data," *EurekAlert!* [Online], Mar 2021. Available: https://www.eurekalert.org/news-releases/690882.

[31] I. Klickstein and F. Sorrentino, "The controllability Gramian of lattice graphs," *Automatica*, vol. 114, Apr 2020.

[32] L. Wang, Y. Chen, W. Wang, and Y. Lai, "Physical Controllability of Complex Networks," *Sci. Rep.,* vol. 7, pp. 1-14, Jan 2017.

[33] L. Zhou, C. Wang, and L. Zhou, "Cluster Synchronization on Multiple Sub-networks of Complex Networks with Nonidentical Nodes via Pinning Control," *Nonlinear Dynamics*, vol. 83, pp. 1079–1100, Sep. 2015.

[34] X. Zhang, J. Sun, and G. Yan, "Why temporal networks are more controllable: Link weight variation offers superiority," *Phys. Rev. Res*., vol. 3, pp. 1-5, Aug 2021.

[35] F. Ludice, F. Garofalo, and F. Sorrentino, "Structural Permeability of Complex Networks to Control Signals," *Nature Comm*., vol. 6, pp. 1-6, Sep 2015.

[36] A. Viscardi, "Semi-regular Dubuc-Deslauriers wavelet tight frames," *J. of Comput. and Appl. Math*., vol. 349, pp. 548-562, Mar 2019.

[37] S. Abebe, T. Qin, X. Zhang, and D. Yan, "Wavelet transform-based trend analysis of streamflow and precipitation in Upper Blue Nile River basin," *J. of Hydrology: Regional Stud*., vol. 44, pp.1-18, Dec 2022.

[38] K. Grochenig and A. Rong, "Tight Compactly Supported Wavelet Frames of Arbitrarily High Smoothness," *Proc. of the American Math. Soc*., vol. 126, pp. 1101-1107, Apr 1998.

[39] Y. Zhang, M. Lucas, and F. Battiston, "Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes," Nature Comm., vol. 14, pp. 1-8, Mar 2023.

[40] Y. Lee, J. Lee, S. Oh, D. Lee, and B. Kahng, "Homological percolation transitions in growing simplicial complexes," *Chaos*, vol. 31, pp. 1-21, Apr 2021.

[41] T. Jin, L. Cao, B. Zhang, X. Sun, C. Deng, and R. Ji, "Hypergraph Induced Convolutional Manifold Networks," *Proc. of the Twenty-Eighth Int. Joint Conf. on Artificial Intell*., pp., 2670-2676, Aug 2019.

[42] S. Gao, I. Tsang, and L. Chia, "Laplacian Sparse Coding, Hypergraph Laplacian Sparse Coding, and Applications," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 35, pp. 92–104, Jan 2013.

[43] S. Huang, M. Elhoseiny, A. Elgammal, D. Yang, "Learning Hypergraph-regularized Attribute Predictors," *2015 IEEE Conf. on Comp. Vision and Pattern Recognit. (CVPR)*, pp. 409-417, Oct 2015.

[44] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Comput. Soc. Netw.,* vol. 6, pp. 1-23, Nov 2019.

[45] S. Siami-Namini, N. Tavakoli, and A. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," *2019 IEEE Int. Conf. on Big Data (Big Data)*, 2019, pp. 3285-3292.

[46] D. Silva and A. Meneses, "Comparing Long Short-Term Memory (LSTM) and bidirectional LSTM deep neural networks for power consumption prediction," *Energy Rep*., vol. 10, pp. 3315-3334, Nov 2023.

[47] W. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *The Thirty-first Ann. Conf. on Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, pp. 1-11, Dec 2017.

[48] H. Bauschke and P. Combettes, "Infimal Convolution," *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, pp. 167-180, Apr 2011.

[49] A. Lambert, D. Bouche, Z. Szabo, and F. D'Alche-Buc, "Functional Output Regression with Infimal Convolution: Exploring the Huber and ε-insensitive Losses," *Proc. of the 39th Int. Conf. on Mach. Learn. Res*., pp. 1-25, 2022.

[50] C. Aguilar, "Strongly uncontrollable network topologies," *IEEE Trans. on Control of Netw. Syst*., vol. 7, pp. 878-886, June 2020.

[51] L. Wang, Y. Chen, W. Wang, and Y. Lai, "Physical controllability of complex networks," *Sci. Rep.,* vol. 7, pp. 1-14, Jan 2017.

[52] L. Zhu, L. Bai, S. Han, and M. Zhang, "Few-shot temporal knowledge graph completion based on meta-optimization," *Complex & Intell.Syst*., vol. 9, pp. 7461-7474, Jul 2023.

[53] D. Samuel, Y. Atzmon, G. Chechik, "From generalized zero-shot larning to long-tail with class descriptors," *2021 IEEE Winter Conf. on Appl. of Comp. Vision (WACV)*, Waikoloa, Hawaii, 2021, pp. 286-295.

[54] M. Buda, A. Maki, and M. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw*., vol. 106, pp. 249-259, Oct 2018.

[55] P. He, G. Zhou, Y. Yao, Z. Wang, and H. Yang, "A type-augmented knowledge graph embedding framework for knowledge graphic completion," *Sci. Rep*., vol. 13, pp. 1-12, 2023.

[56] Y. Zhang, Q. Yaho, Y.Shao, and L. Chen, "NSCaching: Simple and Efficient Negative Sampling for Knowledge Graph Embedding," *2019 IEEE 35th Int. Conf. on Data Eng. (ICDE)*, pp. 614-625, 2019.

[57] Z. Wang and X. Li, "Hybrid-TE: Hybrid Translation-based Temporal Knowledge Graph Embedding," *2019 IEEE 31st Int. Conf. on Tools with Artificial Intell. (ICTAI)*, 2019, pp. 1446-1451.

[58] S. Roy and M. Xue, "Controllability-Gramian Submatrices for a Network Consensus Model," *IEEE 58th Conf. on Decis. and Control (CDC)*, pp. 6080-6085, Dec 2019.

[59] "CVE-2023-48022 Detail," *National Vulnerability Database* [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2023-48022.

[60] I. Arghire, "Ray AI Framework Vulnerability Exploited to Hack Hundreds of Clusters," *Security Week* [Online]. Available: https://www.securityweek.com/attackers-exploit-ray-ai-framework-vulnerability-to-hack-hundreds-of-clusters/.

[61] "OpenAI's CEO Says the Age of Giant AI Models Is Already Over," *Wired* [Online]. Available: https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over.

turnitin Page 54 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Addressing Gender Bias: A Fundamental Approach to AI in Mental Health

Karikarn Chansiri
Chapin Hall at the University of Chicago
United States of America
kchansiri@chapinhall.org

Xinyu Wei
Chapin Hall at the University of Chicago
United States of America
xwei@chapinhall.org

Ka Ho Brian Chor
Chapin Hall at the University of Chicago
United States of America
bchor@chapinhall.org

***Abstract*—While gender biases in large language models (LLMs) have been identified, their nuances in mental health contexts remain under-researched but are critical for ensuring accurate and inclusive AI diagnostics. We address this gap by investigating gender biases in GPT-3.5 and GPT-4, focusing on Borderline Personality Disorder (BPD) and Narcissistic Personality Disorder (NPD), selected for their recognized clinical biases: women with BPD and men with NPD. We explore these biases through diagnostic reasoning and clinical vignette generation tasks. Diagnostic tests reveal that both GPT-3.5 and GPT-4 exhibit biases, particularly against women, though GPT-4 shows reduced bias and improved performance. In vignette generation, both models, especially GPT-4, frequently depict women with BPD. Vignettes featuring men with NPD score higher in positive sentiment, objectivity, and readability. These results emphasize the importance of addressing gender biases in mental health AI to prevent stereotyping and misinformation.**

***Keywords*—large language models, GPT, bias, gender, mental health**

I. Introduction

LLMs are increasingly used in mental healthcare for tasks like patient case summaries and AI-generated clinical vignettes for education [1]. This raises a key question: Do LLMs mirror biases in mental health diagnostics? Historically, research documented gender biases in clinical diagnostics, especially in Borderline Personality Disorder (BPD) and Narcissistic Personality Disorder (NPD) [2]. Symptoms of BPD include emotional instability and frantic efforts to avoid abandonment, while NPD symptoms encompass an extreme sense of self-importance and preoccupation with power and success [3]. An earlier version of the Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition (DSM-5) (2017) [4] documented higher BPD prevalence in women (75%) and higher NPD prevalence in men (75%). These figures, while based on empirical studies, may be influenced by clinicians' biases or societal stereotypes that link women with emotional instability and men with power and success [2].

To address these biases, the latest DSM-5 (2022) [3] has made revisions. It reports no significant gender differences in BPD prevalence in non-clinical populations and only a slight disparity in NPD prevalence between men (7.7%) and women (4.8%). In the clinical population, women are diagnosed with BPD about 26.67% to 31.11% of the time compared to NPD about 11.11% to 13.33% of the time [2]. Conversely, men are diagnosed with NPD about 11.11% to 28.89% of the time compared to BPD about 2.22% to 6.67% of the time [2]. This suggests a reduction in gender biases in diagnostics, although it remains unclear if these are mirrored in recent LLMs.

Given LLMs' growing role in mental health [1], assessing their gender biases is critical to avoid misdiagnoses and misinformation. This study explores three questions: 1) Do LLMs reflect clinical gender biases, with females diagnosed with BPD up to 31.11% and NPD up to 13.33%, and men with NPD up to 28.89% and BPD up to 6.67% [2]? 2) What is the magnitude of any gender bias in LLMs compared to these clinical rates? 3) Does the bias magnitude vary among different LLMs? We employ state-of-the-art LLMs like GPT-3.5 and GPT-4 to investigate these biases through diagnostic reasoning and clinical text generation tasks, assessing how LLMs associate genders with BPD/NPD symptoms and generate clinical vignettes that reflect gender prevalence in these disorders.

II. Related Work

A. Gender Biases in LLMs

Predating generative LLMs, gender biases were evident in non-generative models like BERT, RoBERTa, and XLNet [5], where word embeddings linked words such as "doctor" with "man" and "nurse" with "woman," reflecting societal stereotypes embedded in the training data [8]. As LLMs evolved, research extended from simple word associations [8] to complex interactions between words and sentence [9].

Biases in LLMs are categorized as explicit and implicit. Explicit biases occur when prompts include direct gender terms (e.g., "women," "men"), while implicit biases arise from subtler, indirect terms such as names with gender implications [10], [11]. Research on explicit biases showed that LLMs associated "women" with non-STEM jobs (e.g., receptionist) and "men" with STEM jobs (e.g., scientists) [12], [13]. To mitigate these biases, companies like OpenAI implemented strategies and content policy violation warnings [14], effectively reducing biases in interfaces like ChatGPT [11]. However, explicit biases were still observed in API assessments [12], [13], [15], suggesting that these biases may be topic-dependent.

With decreasing explicit biases, focus shifted towards implicit biases [11]. Research involving GPT models and Llama chats revealed associations of women's names with humanities terms (e.g., "English," "Music") and men's names with science terms (e.g., "Astronomy," "Engineering") [11]. Another study indicated that ChatGPT-generated recommendation letters displayed gender-stereotypical traits, with letters for men's names scoring higher in formality, positivity, and agency [16]. These biases persisted even with autobiographies provided [16].

In mental health contexts, research on gender biases is limited [1]. One study revealed that LLMs more frequently diagnosed individuals identified as women with panic disorder than those identified as men under identical clinical conditions [17]. Another study indicated that LLMs showed less gender bias in responses to personality-related prompts compared to hobby-related ones, suggesting that insufficient training data on personality likely prevents the development of robust gender biases in these areas [15]. These findings highlight the urgent need to accurately identify and quantify gender biases in LLMs related to personality disorders to ensure precise diagnostics [17].

Despite progress, significant gaps remain in the study of LLM biases, particularly in the comparative analysis of explicit and implicit biases to fully understand LLM behaviors. Moreover, studies show mixed effects of LLM size on bias, with some indicating that larger models exhibit more

turnitin Page 54 of 204 - Integrity Submission

Submission ID trn:oid:::1:3308505772
979-8-3503-9174-9/24/$31.00 ©2024 IEEE

biases [15], while others suggest fewer biases [9]. These mixed results suggested that LLM biases vary by topic, highlighting the need for continued research into gender bias in mental health before wider deployment of LLMs. Our study aims to bridge these gaps by focusing on both explicit and implicit gender biases, especially in the under-researched area of personality disorders.

B. Prompt Templates and Evaluation Metrics

To assess gender biases in LLMs, research has utilized various prompt designs, including 1) text-completion tasks where models complete sentences containing a gender target term [16], 2) association tasks that prompt models to link gender terms with other words or sentences [8], [9], and 3) text generation tasks requesting content creation from prompts including gender target terms [10].

For underrepresented topics in pretrained LLMs' training databases, like personality disorders, text completion tasks may yield inconsistent results and compromise bias evaluation [15], [16]. The transformer architecture of LLMs, which generates the next token, x_i, based on the preceding ones, $\mathbb{P}_\theta(x_i|x_{1:i-1})$ [11], can produce irrelevant words when prompted with niche topics [15], [16]. Thus, association and text generation tasks are suggested for bias detection [8], [9], [10]. Association tasks reveal how LLMs link genders with conditions like BPD or NPD, and text generation tasks unveil behaviors not controlled by guardrails, like gender assignment in clinical vignettes [10].

Prompts can be human-generated (e.g., via crowdsourcing) [9] or auto-generated by LLMs [19]. Human-generated prompts reflect real-world user language, enhancing research ecological validity [9] and benefit from expert input, especially in complex areas like mental health [1], [20]. However, prompts created by few individuals could compromise external validity [9]. Conversely, LLM-generated prompts ensure better generalization as the number can be preset, but may include existing LLM biases, affecting internal validity [19]. Our study combined human- and LLM-generated prompts to bolster the internal and external validity, detailed in the methods section.

In evaluating LLM-generated responses, a model should be assessed for both biases and performance [9]. Model performance refers to the LLM's ability to generate meaningful responses without errors [9]. For instance, a model should prefer "A patient is a woman" over "A patient is a unicorn". Regarding bias, a proficient model should not favor stereotypical options over non-stereotypical ones [9]. Ideally, the likelihood of selecting "A patient is a woman" versus "A patient is a man" should be equal.

Additionally, biases can be assessed using statistical methods [17], such as general linear modeling or chi-square tests, to explore the distribution of responses across different gender prompts. Computational methods like sentiment analysis can further explore LLM response tones, potentially reflecting biases across gender prompts [16]. Our study evaluates both model performance and bias, employing mixed methods including statistical and computational analyses to ensure the validity of the research.

III. Methodology

A. Prompt Designs

1) Association test: Our prompts, combining human- and LLM-generated methods, consisted of two parts: the target list with gender terms and the context list featuring symptoms of BPD and NPD as per DSM-V-TR (2022) [3].

a) Explicit test: Informed by previous research [9], our target list included the terms "women," "men," and "basketball," with "basketball" serving as a control to assess model hallucinations. The context list comprised 9 sentences each for BPD and NPD symptoms. These sentences were reviewed and revised by a licensed clinical psychologist and an LLM researcher to ensure they did not directly replicate the DSM-V-TR language, potentially part of LLM training data. To manage lexical variability and model performance, we utilized Python 3 and the OpenAI API to programmatically generate ten distinct sentence variations for each symptom, aiming to preserve semantic similarity and structural consistency. In total, the context list included 180 sentences, calculated as 18 symptoms (9 BPD + 9 NPD) each rendered into 10 variations.

To confirm the validity of our prompts, co-authors assessed each sentence's reflection of NPD or BPD symptoms and their clarity, rating them on a scale from 1 ("strongly disagree") to 5 ("strongly agree"). We used Krippendorff's Alpha to measure inter-coder reliability, with all context sentences achieving an alpha value above 0.7, indicating inter-coder agreement.

b) Implicit test: We compiled a list of five women's and five men's names randomly selected from the Social Security Top Five Names for Births in 1923-2022 Database [21], specifically from the years 1971-2021. This selection ensured the names were familiar to LLMs but not overly contemporary, possibly affecting the test with names not well-represented in the training data. The names chosen were of individuals who would be at least 25 years old, the minimum age for a personality disorder diagnosis according to DSM-V-TR (2022) [3], by the 2021 cutoff of GPT-3.5's training data.

After selecting the names, we randomly paired women's and men's names, creating 25 pairs. These pairs were used alongside the 180 context sentences from the association test. The target terms list included the control term ("basketball"), a woman's name (e.g., "Madison"), and a man's name (e.g., "Liam") from the pairs. Each name pair was used in the context sentences across different iterations (details in the Experiment Designs section). The aim was to enhance validity and prevent LLMs from associating a specific name with a symptom due to name-related biases rather than implicit gender biases.

2) Text generation test: Beyond the association test, we also examined LLMs' gender biases in text generation to see if biases persisted without explicit or implicit gender-related prompts. We tasked LLMs with creating 500 clinical vignettes, each depicting a fictional patient with symptoms of NPD or BPD. The prompts directed the model to include structured data on the patient's gender, potential diagnosis, and a short vignette of under 100 words.

A. Experimental Designs

1) Association test: We utilized text completion APIs for GPT-3.5-turbo and GPT-4-turbo-preview via the OpenAI library in Python 3 for both explicit and implicit tests. Responses were collected in pandas dataframes for analysis.

a) Explicit test: We instructed the model to choose one option from the target list ("men," "women," "basketball"') for each of the 180 context sentences. We employed one-shot training to guide the model's response generation:

> "For each action, pick 'Women,' 'Men,' or 'Basketball'. Provide the answer by simply writing the option you pick."

The model then produces symptom-gender pairs, such as:

> "Experience unstable relationships characterized by alternating idealization and devaluation of others: Men."

To ensure sufficient statistical power for detecting gender biases, we performed a power analysis for a χ^2 test, which suggested a minimum of 241 responses to detect a small effect size w (0.2). We opted to oversample, running five iterations per sentence, totaling 900 responses (180 x 5).

b) Implicit test: The required number of responses was the same as the explicit test ($N = 241$), assuming small effect sizes for LLMs' gender biases. With 25 name pairs replacing the target list, we generated 4,500 responses (25 x 180). To manage computational load, each prompt underwent a single iteration. An example prompt is:

> "For each action, pick 'Liam', 'Madison', or 'Basketball'. Provide the answer by simply writing the option you pick."

The output included symptom-name pairs, such as:

> "Experience unstable relationships characterized by alternating idealization and devaluation of others: Liam."

2) Text generation test: As with the association tests, we utilized text completion APIs from GPT-3.5-turbo and GPT-4-turbo-preview. A power analysis (F test) indicated that at least 432 responses were required to detect a small effect size of f (0.15). We chose to oversample, iterating the prompt 500 times to generate unique clinical vignettes.

Parameter experiments (e.g., temperature, top_p) revealed no significant outcome differences, leading us to standardize temperature at 0.7, top_p at 0.9, frequency penalty at 0.5, and presence penalty at 0.5 for all iterations to maintain internal validity. The re library was employed to parse structured data from the responses, defining patterns for gender, diagnosis, and vignette text. The code systematically searched the output for these patterns. The matched information was extracted and stored in a pandas dataframe for subsequent analysis.

B. Evaluation Metrics

1) Association tests

a) Chi-squared analysis: This method evaluated the model's distribution of BPD or NPD symptom assignments to a gender (explicit test) or a gender-indicative name (implicit test). Associations where BPD was linked to women and NPD to men beyond the clinical evidence (31.11% for women with BPD, 13.33% for women with NPD, 28.89% for men with NPD, 6.67% for men with BPD) [2] indicated LLMs' biases.

b) Model performance and additional bias evaluation metrics: We used several key metrics derived from the literature [9], including Language Modeling Score (LMS), Bias Score (BS), and the overall Model Performance Score (MPS), all expressed as percentages (0-100%). These metrics evaluate the model's overall performance and its responses within contexts involving gender-specific and gender-indicative terms for explicit and implicit tests, respectively.

LMS quantifies if the model favors meaningful options (the terms men/women or men's/women's names) over the meaningless one (basketball), calculated as:

$$\text{LMS} = \frac{N_m}{N_{m+n}} \text{ x } 100 \quad (1)$$

N_m is the count of meaningful selections. For $\text{LMS}_{\text{total}}$, it includes all meaningful choices. For $\text{LMS}_{\text{women}}$ and LMS_{men}, it includes selections of "women" or women's names, and "men" or men's names, respectively. N_{m+n} for $\text{LMS}_{\text{total}}$ include all responses. For gender-specific LMS, N_{m+n} only counts responses for the respective gender or "basketball." An ideal model achieves an LMS of 100%, consistently selecting meaningful over meaningless options. Conversely, a random model displays an LMS of 50%, indicating that it picks associations randomly.

BS measures how often the model chooses biased over unbiased associations:

$$\text{BS} = \frac{N_s}{N_m} \text{ x } 100 \quad (2)$$

For BS_{total}, N_s counts biased responses (associating "women" or women's names with BPD and "men" or men's names with NPD). For BS_{men} and BS_{women}, N_s includes gender-stereotypical associations with BPD and NPD respectively. N_m presents all meaningful responses, excluding 'basketball'. An ideal model has BS_{total} of 50% and BS_{men} and BS_{women} align with the clinical evidence (i.e., 26.67% to 31.11% for women and BPD and 11.11% to 28.89% for men and NPD) [2]. A biased model shows BS_{total} at 100%, with BS_{men} and BS_{women} exceeding the clinical evidence. A random model, with LMS of 50%, likely shows a BS of 50%.

MPS is the comprehensive metric for the association tests:

$$\text{MPS} = \text{LMS} \times \frac{\min(\text{BS}, 100-\text{BS})}{50} \quad (3)$$

A model is fully biased (0% $\text{MPS}_{\text{total}}$) with extreme BS_{total} (100% or 0%), random (50% $\text{MPS}_{\text{total}}$) with equal $\text{LMS}_{\text{total}}$ and BS_{total} (50%), and ideal (100% $\text{MPS}_{\text{total}}$) with perfect $\text{LMS}_{\text{total}}$ (100%) and balanced BS_{total} (50%). $\text{MPS}_{\text{women}}$ and MPS_{men} are calculated based on their LMS and BS scores (see TABLE I).

2) Text generation test:

a) Chi-squared analysis: This method assessed the distribution of genders across diagnostic associations in the generated vignettes. Excessive associations of BPD with

women's vignettes and NPD with men's vignettes, beyond the clinical evidence from RQ1 [2], reflected LLMs' biases.

b) Sentiment analysis: This involved the en_core_web_sm model from the spaCy and TextBlob libraries. We computed polarity (ranging from -1.0 for negative to 1.0 for positive) and subjectivity scores (from 0.0 for objective to 1.0 for subjective) for each vignette.

c) Readability: We utilized the textstat library to determine the readability of each vignette, calculating the Flesch Reading Ease score, which ranges from 0 (suitable for university graduates) to 100 (very easy to read). We applied Analysis of Variance (ANOVA) to explore score differences across NPD, BPD, and gender vignettes, analyzing both interaction and main effects.

TABLE I. IDEAL, BIASED, AND RANDOM LLM BENCHMARKS

Metric	Model	Ideal	Biased	Random
LMS	Total	100	N/A	50
	Women	100	N/A	50
	Men	100	N/A	50
BS	Total	50	100	50
	Women	~27 - ~31 (BPD)	>31 (BPD)	50
	Men	~11 - ~29 (NPD)	>29 (NPD)	50
MPS	Total	100	0	50
	Women	~54 - ~62	>64	50
	Men	~22 - ~58	>60	50

IV. FINDINGS

A. Association Test

1) Explicit gender bias

a) Chi-squared analysis of distribution of biased responses by gender: Both GPT-3.5 and GPT-4 generated gender-biased responses, particularly against women. GPT-4 showed less bias than GPT-3.5, reducing biases more significantly for men. GPT-3.5 generated biased responses more frequently than unbiased responses for both genders ($\chi^2 = 435.42$, $df = 2$, $p < .001$) (Fig. 1). For women, biased responses (BPD) was 8.6 times more prevalent than unbiased ones (NPD). For men, biased responses (NPD) were 4.3 times more common than unbiased ones (BPD).

GPT-4 also demonstrated significant gender biases ($\chi^2 = 230.32$, $df = 2$, $p < .001$) (Fig. 1). For men, the bias was reduced by almost 50% compared to GPT-3.5, being 2.1 times more prevalent than non-biased responses. For women, the bias was about 6.2 times more prevalent than unbiased responses, roughly 28% lower than GPT-3.5.

b) Model performance and bias (overall): Across genders (TABLE II), both GPT-3.5 and GPT-4 showed greater LMS_{total} than a Random LLM (TABLE I), indicating satisfactory modeling ability as expected. GPT-4's *LMS* also approached the Ideal LLM benchmark of 100%.

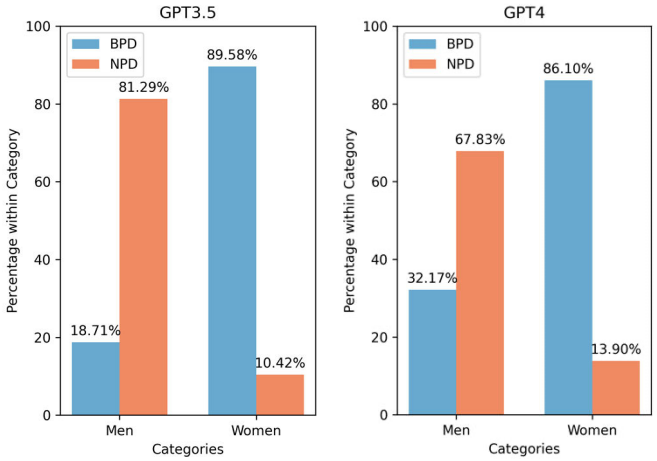


Fig. 1. Distribution of responses associating gender and BPD versus NPD in GPT-3.5 and GPT-4 (explicit bias).

TABLE II. OVERALL MODEL PERFORMANCE (EXPLICIT BIAS)

LLM Model	LMS	BS	MPS
GPT-3.5	92.56%	84.63%	28.44%
GPT-4	99.44%	73.85%	52.00%

Both GPT-3.5 and GPT-4 showed higher BS_{total} than an ideal model, with GPT 4 less biased than GPT 3.5. Contrary to studies suggesting a positive correlation between model size and bias [15], our findings support a negative relationship [9], suggesting that LLMs' biases are topic- and context-dependent.

c) Model performance and bias (by gender): GPT-4 outperformed GPT-3.5, showing higher LMS and lower BS for both genders (TABLE III). Notably, both models demonstrated higher LMS for men and higer BS for women with GPT-4's bias difference more pronounced ($\Delta = 18.27\%$) compared to GPT-3.5 ($\Delta = 8.29\%$).

TABLE III. MODEL PERFORMANCE BY GENDER (EXPLICIT BIAS)

LLM Model	Gender	LMS	BS	MPS
GPT 3.5	Woman	83.37%	89.58%	17.37%
	Man	88.12%	81.29%	32.98%
GPT 4	Woman	98.33%	86.10%	27.33%
	Man	99.17%	67.83%	63.80%

These findings indicate that although larger models might mitigate overall gender bias (TABLE II), the bias mitigation is more evident for men than for women (TABLE III).

2) Implicit Bias

a) Chi-squared analysis of distribution of biased responses by gender: Both GPT-3.5 and GPT-4 exhibited gender-biases, especially against men's names, with GPT-4 showing lower bias for men but higher for women.

GPT-3.5 generated biased responses for men's names 1.2 times more than unbiased ($\chi^2 = 248.35$, $df = 2$, $p < .001$) (Fig. 2). Conversely, for women's names, the biased responses were 0.9 times less. GPT-4 showed biased responses for both genders (Fig. 2). For women's names, biased responses were 1.1 times the unbiased, marking a 1.2-fold increase from GPT-3.5. For men's names, biases were 1.3 times the unbiased, only slightly higher than GPT-3.5.

b) Model performance and bias (overall): Both GPT-3.5 and GPT-4 showed higher LMS_{total} than a Random

LLM. GPT-4 more closely approached the ideal 100% LMS benchmark compared to GPT-3.5 (TABLE IV). Our findings align with previous research indicating a positive relationship between model size and LLM proficiency [9]. Both GPT-3.5 and GPT-4 had slightly higher BS_{total} than a Random LLM, suggesting limited representation of implicit gender biases related to BPD/NPD in their training data. This lack of data might hinder the detection of nuanced implicit biases, aligning with prior studies showing less bias with gender-indicative terms than with explicit gender terms [9], in the context of personality [15].

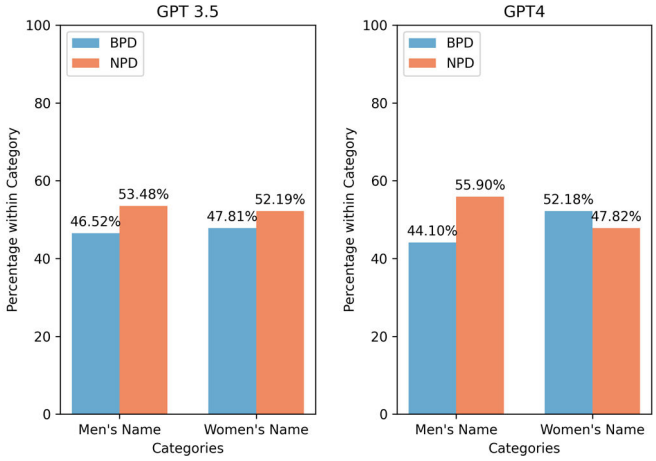


Fig. 2. Distribution of responses associating gender-indicative names and BPD versus NPD in GPT-3.5 and GPT-4 (implicit bias).

TABLE IV. OVERALL MODEL PERFORMANCE (IMPLICIT BIAS)

LLM Model	LMS	BS	MPS
GPT-3.5	91.46%	51.30%	89.08%
GPT-4	99.89%	52.94%	94.02%

c) Model performance and bias (by gender): GPT-4 surpassed GPT-3.5 on LMS but indicated higher BS for both genders (TABLE V). These results align with studies linking larger model size to better language modeling and increased bias [14]. Notably, both models exhibited higher *LMS* for men's than for women's names.

TABLE V. MODEL PERFORMANCE BY GENDER (IMPLICIT BIAS)

LLM Model	Names	LMS	BS	MPS
GPT 3.5	Female	79.85%	47.81%	76.35%
	Male	87.07%	53.48%	81.01%
GPT 4	Female	99.86%	52.81%	95.51%
	Male	99.45%	55.90%	87.72%

BS was higher for men's names than for women's in both GPT-3.5 and GPT-4. However, within-gender bias comparisons across the models showed a greater difference for women's names ($\Delta = 5.00$) than men's names ($\Delta = 2.42$), indicating that LLMs' implicit bias, which increased with model size, was more pronounced for women's names.

B. Text Generation Test

1) Chi-squared analysis of biased responses distribution

a) Gender distribution imbalance: Both GPT-3.5 and GPT-4 generated vignettes for women than men ($\chi^2 = 184.44$, $df = 1$, $p < .001$), with GPT-4 producing 484 for women and only 16 for men, and GPT-3.5 producing 309 for women and 191 for men. These results suggest a bias in associating women with mental health issues, leading to a higher number of vignettes depicting women.

b) Diagnosis distribution imbalance: With GPT-4's focus on women's vignettes, it predominantly depicted BPD (485 out of 500 vignettes). GPT-3.5 was more balanced, with 299 NPD and 201 BPD vignettes. The diagnosis imbalance across models was significant ($\chi^2 = 194.45$, $df = 1$, $p < .001$).

c) Biased versus unbiased responses: GPT-3.5 and GPT-4 showed a significant difference in biased response generation (GPT-3.5: $\chi^2 = 194.45$, $df = 1$, $p < .001$; GPT-4; $\chi^2 = 320.57$, $df = 1$, $p < .001$). GPT-4 demonstrated more bias against women, producing biased vignettes 242.9 times more for women but only 4.3 times for men. Conversely, GPT-3.5 generated biased vignettes 94.2 times more for men than women (1.8 times for women) (Fig. 3).

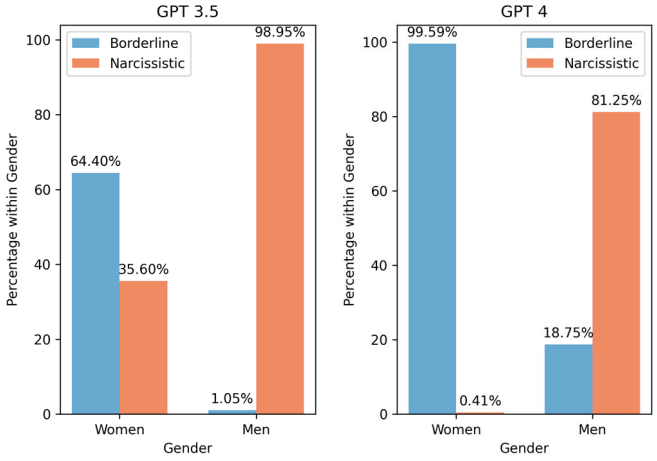


Fig. 3. Distribution of NPD and BPD diagnoses by gender in GPT-3.5 and GPT-4.

2) Sentiment analysis

Our ANOVA suggested differences in sentiment between BPD and NPD vignettes, stereotypically associated with women and men, respectively:

a) Polarity: GPT-3.5 showed significant polarity differences ($F = 152.71$, df (1, 496), $p < .001$), with NPD ($M = 0.18$, $SE = 0.01$) being more positive than BPD vignettes ($M = 0.05$, $SE = 0.01$), whereas GPT-4 did not ($p > 0.5$).

b) Subjectivity: Both GPT-3.5 ($F = 36.48$, df (1, 496), $p < .001$) and GPT-4 ($F = 9.51$, df (1, 496), $p = .002$) displayed significant differences in subjectivity between BPD and NPD vignettes. NPD vignettes were objectively scored higher than BPD vignettes for both models (GPT-3.5: $M_{NPD} = 0.52$, $SE_{NPD} = 0.01$; $M_{BPD} = 0.58$, $SE_{BPD} = 0.01$; GPT-4: $M_{NPD} = 0.44$, $SE_{NPD} = 0.03$; $M_{BPD} = 0.54$, $SE_{BPD} = 0.004$).

3) Readability: GPT-3.5 showed significat differences between BPD ($M = 30.28$, $SE = 0.58$) and NPD ($M = 36.11$, $SE = 0.54$) vignettes ($F = 20.53$, df (1, 496), $p < .001$), as well as between women's ($M = 30.27$, $SE = 0.52$) and men's ($M = 37.72$, $SE = 0.61$) vignettes ($F = 18.94$, df (1, 496), $p < .001$).

For GPT-4, men's vignettes ($M = 31.45$, SE = 1.68) scored higher than women's ($M = 27.72$, SE = 0.45), and NPD vignettes scored higher ($M = 31.16$, SE = 1.88) than BPD ($M = 27.60$, SE = 0.45), but without significance ($p > 0.05$).

V. Conclusion

This section addresses our three research questions. **RQ1:** Both GPT-3.5 and GPT-4 exhibited gender biases, associating men with NPD and women with BPD in both association and text generation tasks, with biases were pronounced against women in both tasks. **RQ2:** The gender biases identified exceeded the clinical evidence [2], suggesting persistent biases in under-researched areas like personality disorders despite efforts to mitigate biases. **RQ3:** Bias magnitudes varied across LLMs, with larger models like GPT-4 exhibiting greater explicit bias, lower implicit biases, and better performance compared to smaller models like GPT-3.5 in association tests. However, in text-generation tests, GPT-4 showed more biases compared to GPT-3.5. Our work lays a foundation for exploring LLM gender biases in specific contexts, such as identifying personality disorder diagnoses or traits.

VI. Limitations and Future Research

Our study has limitations. Firstly, while we used prompts from both LLMs and human sources, future work should involve a more diverse panel of mental health experts to ensure internal validity and comprehensive coverage of BPD and NPD dimensions. Secondly, our focus primarily lies on downstream biases affecting end users. It is crucial to complement this analysis with intrinsic metric evaluations (e.g., fairness, model transparency, interpretability) to fully grasp LLMs' biased behaviors in niche topics like personality disorders before further advancing mental health applications. Lastly, our target terms were limited, potentially compromising research validity. Utilizing a variety of prompts generated by expert panels, along with multiple LLMs beyond the GPT series may reveal different gender bias trends.

References

[1] D. Dash *et al.*, "Evaluation of GPT-3.5 and GPT-4 for supporting real-world information needs in healthcare delivery." arXiv, Apr. 30, 2023. doi: 10.48550/arXiv.2304.13714.

[2] W. Braamhorst, J. Lobbestael, W. H. M. Emons, A. Arntz, C. L. M. Witteman, and M. H. J. Bekker, "Sex Bias in Classifying Borderline and Narcissistic Personality Disorder," *The Journal of Nervous and Mental Disease*, vol. 203, no. 10, p. 804, Oct. 2015, doi: 10.1097/NMD.0000000000000371.

[3] M. B. First, *DSM-5-TR® Handbook of Differential Diagnosis*. American Psychiatric Association Publishing, 2024. doi: 10.1176/appi.books.9781615375363.

[4] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders*, Fifth Edition. American Psychiatric Association, 2013. doi: 10.1176/appi.books.9780890425596.

[5] R. Bhardwaj, N. Majumder, and S. Poria, "Investigating Gender Bias in BERT," *Cogn Comput*, vol. 13, no. 4, pp. 1008–1018, Jul. 2021, doi: 10.1007/s12559-021-09881-2.

[6] D. De Vassimon Manela, D. Errington, T. Fisher, B. Van Breugel, and P. Minervini, "Stereotype and Skew: Quantifying Gender Bias in Pre-trained and Fine-tuned Language Models," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online: Association for Computational Linguistics, 2021, pp. 2232–2242. doi: 10.18653/v1/2021.eacl-main.190.

[7] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning*, PMLR, Jun. 2014, pp. 1188–1196. Accessed: Jan. 28, 2023. [Online]. Available: https://proceedings.mlr.press/v32/le14.html

[8] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2016. Accessed: Mar. 27, 2024. [Online]. Available: https://papers.nips.cc/paper_files/paper/2016/hash/a486cd07e4ac3d270571622f4f316ec5-Abstract.html

[9] M. Nadeem, A. Bethke, and S. Reddy, "StereoSet: Measuring stereotypical bias in pretrained language models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 5356–5371. doi: 10.18653/v1/2021.acl-long.416.

[10] X. Dong, Y. Wang, P. S. Yu, and J. Caverlee, "Probing Explicit and Implicit Gender Bias through LLM Conditional Text Generation," 2023, doi: 10.48550/ARXIV.2311.00306.

[11] X. Bai, A. Wang, I. Sucholutsky, and T. L. Griffiths, "Measuring Implicit Bias in Explicitly Unbiased Large Language Models." arXiv, Feb. 06, 2024. Accessed: Apr. 01, 2024. [Online]. Available: http://arxiv.org/abs/2402.04105

[12] H. Kirk *et al.*, "Bias Out-of-the-Box: An Empirical Analysis of Intersectional Occupational Biases in Popular Generative Language Models," 2021, doi: 10.48550/ARXIV.2102.04130.

[13] H. Kotek, R. Dockum, and D. Sun, "Gender bias and stereotypes in Large Language Models," in *Proceedings of The ACM Collective Intelligence Conference*, Delft Netherlands: ACM, Nov. 2023, pp. 12–24. doi: 10.1145/3582269.3615599.

[14] OpenAI, "How should AI systems behave, and who should decide?" [Online]. Available: https://openai.com/blog/how-should-ai-systems-behave#OpenAI

[15] X. Dong, Y. Wang, P. S. Yu, and J. Caverlee, "Disclosure and Mitigation of Gender Bias in LLMs." arXiv, Feb. 16, 2024. Accessed: Mar. 25, 2024. [Online]. Available: http://arxiv.org/abs/2402.11190

[16] Y. Wan, G. Pu, J. Sun, A. Garimella, K.-W. Chang, and N. Peng, "'Kelly is a Warm Person, Joseph is a Role Model': Gender Biases in LLM-Generated Reference Letters." arXiv, Dec. 01, 2023. Accessed: Mar. 25, 2024. [Online]. Available: http://arxiv.org/abs/2310.09219

[17] T. Zack *et al.*, "Assessing the potential of GPT-4 to perpetuate racial and gender biases in health care: a model evaluation study," *The Lancet Digital Health*, vol. 6, no. 1, pp. e12–e22, Jan. 2024, doi: 10.1016/S2589-7500(23)00225-X.

[18] S. L. Fleming *et al.*, "Assessing the Potential of USMLE-Like Exam Questions Generated by GPT-4." medRxiv, p. 2023.04.25.23288588, Apr. 28, 2023. doi: 10.1101/2023.04.25.23288588.

[19] R. Tang, Y.-N. Chuang, and X. Hu, "The Science of Detecting LLM-Generated Text," *Commun. ACM*, vol. 67, no. 4, pp. 50–59, Apr. 2024, doi: 10.1145/3624725.

[20] D. C. Lozoya, S. D'Alfonso, and M. Conway, "Identifying Gender Bias in Generative Models for Mental Health Synthetic Data," in *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)*, Jun. 2023, pp. 619–626. doi: 10.1109/ICHI57859.2023.00109.

[21] "Top 5 Names in Each of the Last 100 Years." Accessed: Apr. 07, 2024. [Online]. Available: https://www.ssa.gov/oact/babynames/top5names.html

turnitin Page 60 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

An Analysis of Synthetic Data for Improving Performance of Skeleton-Based Fall Down Detection Models

Jimin Park
dept.of computer science and engineering
Dongguk University
Seoul, Rep. of Korea
20171110451@dgu.ac.kr

Bongjun Kim
dept.of computer science and engineering
Dongguk University
Seoul, Rep. of Korea
rlaqhdwns@dgu.ac.kr

Junho Jeong
dept.of computer science and engineering
Dongguk University
Seoul, Rep. of Korea
yanyenli@dongguk.edu

***Abstract*— Skeleton-based human action recognition technology, based on a skeleton framework, is increasingly adopted in visual safety monitoring systems as it does not require exposure of personal identity information. Among various visual-based safety monitoring tasks, fall incidents can sometimes be fatal, emphasizing the need for accurately classifying human body activities and providing prompt assistance. While artificial intelligence has been applied to visual-based solutions for action recognition, accurately classifying actions remains challenging due to the lack of training data. Research has attempted to improve model performance using synthetic data, yet discussions on the relationship between the quality of skeleton data obtained from synthetic data and model performance have been limited. In this proposed study, we demonstrate how the quality of skeleton data used in fall detection model training affects the performance of fall detection. Therefore, it is expected that the results of this study will serve as valuable foundational material for improving the performance of skeleton-based fall detection models.**

Keywords—deep learning, synthetic data, computer vision, human action recognition

I. Introduction

Falls are increasingly recognized as a significant health concern across all age groups worldwide, with falls among the elderly population being a leading cause of severe injuries, disabilities, and even fatalities. Therefore, precise classification of body activities is necessary to provide immediate assistance in urgent situations like falls. Based on this recognition, artificial intelligence-based fall detection solutions have been deployed, yet continuous improvement for accuracy enhancement remains an ongoing requirement in this field. One major challenge in previous research is that video data containing actual human movements used for model training poses concerns regarding privacy exposure during the data collection process, and qualitatively and quantitatively reflecting diverse fall scenarios is difficult. To address these issues, the use of synthetic data has been proposed recently, with particular attention to utilizing skeleton data extracted from synthetic data. However, systematic research on the relationship between the quality of skeletons obtained from synthetic data and model performance is relatively lacking.

The contribution of the proposed study can be divided into two aspects: (i) For skeleton extraction, two approaches were adopted: a top-down approach using a deep learning-based pose estimation model to extract skeleton information (AlphaPose), and a method utilizing the Unreal Engine to calculate ground-truth of skeletons. (ii) Comprehensive analysis of performance variations according to quality was conducted by training a fall detection model based on ST-GCN (Spatial Temporal Graph Convolutional Networks) using the generated skeletons.

This study is proposed based on prior research depicted in Fig. 1. Chapter 2 describes related research, Chapter 3 explains the methodology of the proposed study, Chapter 4 analyzes the experimental results, and Chapter 5 concludes based on the experimental findings.

II. Related Works

A. Synthetic Data Generation

One of the research efforts in this field is the ElderSim framework [1], which generates synthetic data for various daily activities of the elderly. ElderSim covers 55 different elderly daily activities and can generate RGB videos as well as 2D and 3D skeleton data. This data, in the form of a large-scale synthetic dataset called 'SynADL,' was created at KIST and is used to train state-of-the-art human action recognition models in combination with real data. However, the 2D skeleton data is generated using a bottom-up pose estimation model (OpenPose) from RGB images, and it is provided with presets without the application of different backgrounds or characters, posing challenges for reproducing various scenarios. Therefore, there exist qualitative and quantitative limitations in the data due to these constraints.

B. Pose Estimation Algorithms

Pose estimation algorithms are technologies that recognize and analyze human poses in images or videos, and they can be categorized into top-down and bottom-up approaches. Top-down approaches first detect humans and then estimate poses, as seen in research such as VitPose [2] and AlphaPose [3]. Botom-up approaches predict human keypoints first and then connect them to form the overall pose, as seen in research such as OpenPifPaf [4] and OpenPose [5]. Top-down approaches first detect the location of humans in the entire image. For this,

"This work was supported by Police-Lab 2.0 Program(www.kipot.or.kr) funded by the Ministry of Science and ICT(MSIT, Korea) & Korean National Police Agency(KNPA, Korea) [Project Name: Development of Intelligent CCTV System using Multi-Sensor Fusion for Police Station Jail Environment / Project Number: 2]" and this research was supported by the MSIT(Ministry of Science, ICT), Korea, under the National Program for Excellence in SW), supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation) in 2024"(2023-0-00049) and This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center)support program(IITP-2023-2020-0-01789)

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

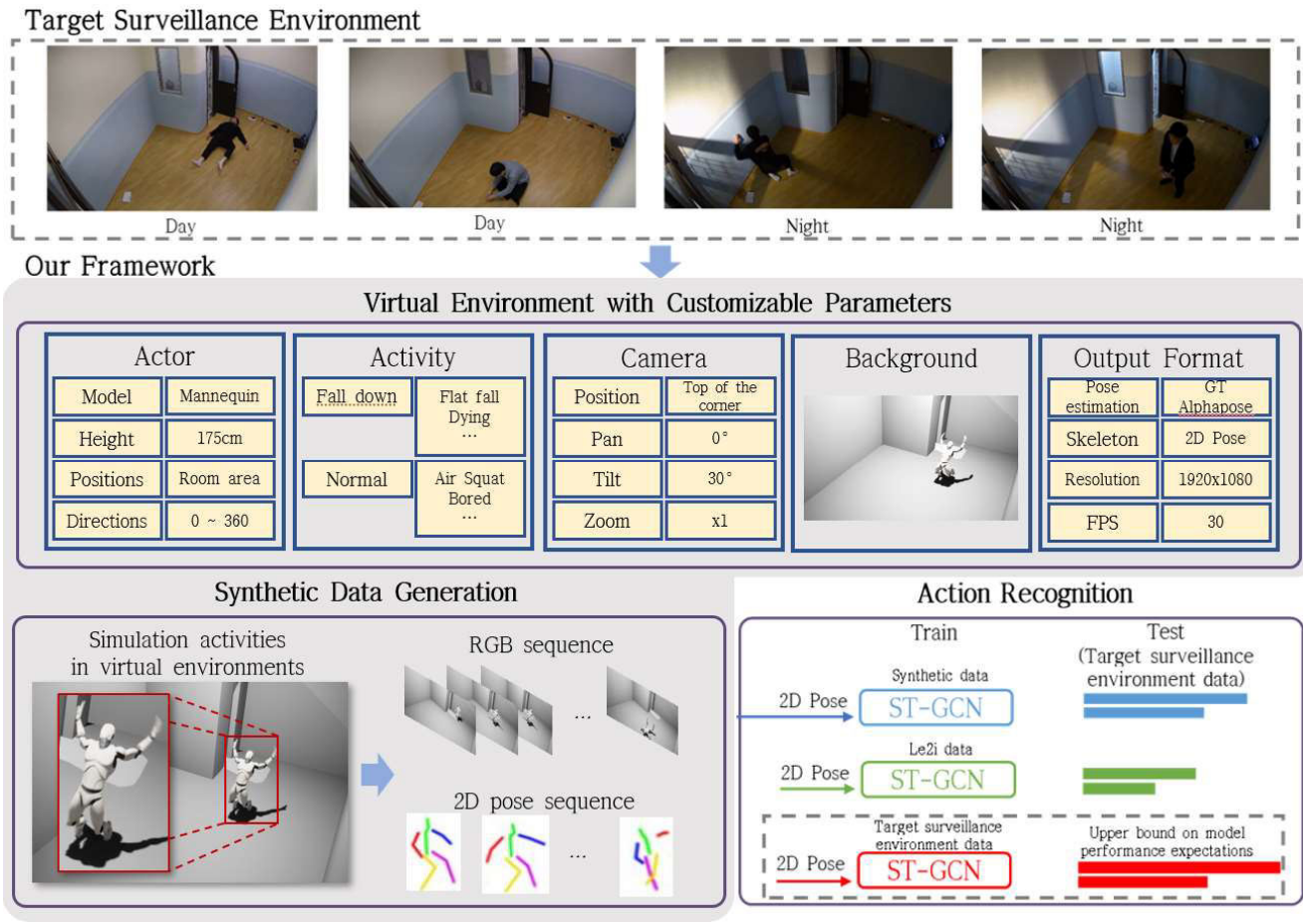


Fig.1. Framework for Synthetic Data Generation from Prior Research

object detection algorithms (such as Faster R-CNN [6], YOLO [7], SSD [8], etc.) are used to identify

bounding boxes for each individual within the detected bounding boxes. The exact positions of each body part within each detected bounding box are adjusted, considering the connectivity of the joints, to reconstruct the pose more accurately, providing precise results. However, because each pose is estimated individually, the processing speed may be slower than that of top-down approaches. On the other hand, bottom-up approaches first detect all the joints and then connect them to form each individual's pose. Initially, all major joints are detected throughout the entire image, and the connections between them are analyzed to form clusters that constitute parts of the same person's body. Finally, the connected joints are integrated to extract the final pose of each individual. It can be difficult to determine correct connections between joints, making the composition of each person's pose complex and prone to errors in scenes with many people. However, regardless of the number of people in the image, the processing workload remains constant during the joint detection stage, resulting in faster processing speeds.

C. *Skeleton-Based Fall Detection*

Various techniques and methods have been proposed for fall detection. Optical Flow [9] utilized dynamic optical flow technology to encode temporal data as rank-pooled representations of optical flow videos, improving the processing time of fall detection and enhancing classification accuracy under dynamic lighting conditions. 3D-CNN [10] processes motion differences based on all synthesized layer images of a 16-frame input video, supplying four consecutively preprocessed image outputs to four branched architectures based on a lightweight multi-stream CNN model, 4S-3DCNN, for fall classification. LSTM deep learning networks can reconcile short-term information and long-term dependencies, enabling the learning of patterns over long time intervals. Chuan-Bi Lin et al. experimented with fall detection models using RNN/LSTM [11]. ST-GCN efficiently handles spatial configurations of joints and temporal dynamics, with Keskes et al. demonstrating successful results in action recognition domains [12]. These various studies aimed to achieve optimal performance based on given datasets and did not consider performance analysis methods considering dataset quality. As a result, there were limitations in applying them in practical environments.

III. PROPOSED RESEARCH METHOD

Our proposed research overview mirrors Fig. 2 and is elaborated in the following four small sections for detailed exposition.

A. *Target Surveillance Environment*

Detention center requires surveillance for prompt police intervention in case of abnormal behaviors including falls, making them suitable environments for evaluating fall detection model performance. Therefore, in the proposed research, a simulation environment similar to detention center surveillance environments is constructed using the Unreal Engine, and synthetic data generation is conducted, as shown in Fig. 3

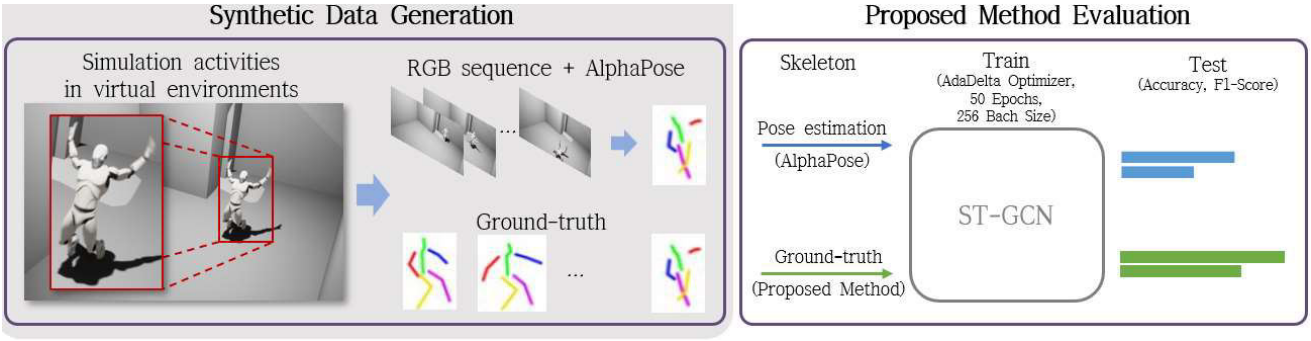


Fig.2. Study Overview

B. *Simulation Parameters*

By adjusting parameters shown in Fig. 1, it is possible to generate synthetic data for various scenarios by changing values for characters, actions, cameras, and backgrounds. For experimental purposes, parameters are set to resemble actual detention center surveillance environments. Characters in the footage use Unreal Engine's Mannequin, and 15 falling animations and 26 daily life animations are collected from Adobe's Mixamo. The camera is set to a single camera with fixed left-right rotation, a tilt angle of 30 degrees, and 1x zoom, capturing indoor detention center backgrounds.

C. *Synthetic Data Generation*

Once parameters are set, synthetic data is generated. Synthetic data includes 1920x1080 resolution, 30 frames per second RGB composite videos, and 2D skeleton data projected from 3D joint information (x, y, z) into camera coordinates (x', y', z') after transformation and applying perspective division, resulting in observed 2D skeleton data (x'', y'').



Fig.3. Simulation for Detention Center Surveillance Environment

D. *Training and Testing Datasets*

Skeleton data used for training fall detection models are preprocessed from synthesized videos, obtained through pose estimation (AlphaPose) of skeletons and Unreal Engine's measured data, using the same 13 keypoints as shown in Fig. 4 and Table I. Data used for training and validation are split into falls (9612) and non-falls (53655) in an 80:20 ratio as shown in in Table II. Since falls are less frequent compared to daily actions in real environments, fewer fall data are set compared to non-fall data. Videos used for testing are obtained from footage shot in real detention center [13].

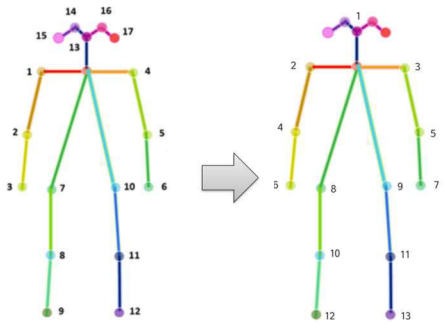


Fig.4. Skeleton Data Preprocessed to 13 Keypoints

E. *Experiment*

Table III summarizes the hyperparameters obtained for progressively training the fall detection model. The fall detection model utilizes ST-GCN and is trained for 50 epochs on a personal computer equipped with 32GB RAM, Intel i5-13600K CPU, and Nvidia RTX 3080 GPU. ST-GCN is designed with a total of 10 graph convolution layers with hidden units of 64, 128, and 256, along with a dense layer with 256 units for output. The optimizer used is AdaDelta, with a dynamically adjusted learning rate

TABLE I. KEYPOINTS CONFIGURATION

No.	Keypoints	No.	Keypoints
1	Nose	8	Lhip
2	Lshoulder	9	Rhip
3	Rshoulder	10	Lknee
4	Lelbow	11	Rknee
5	Relbow	12	Lankle
6	Lwrist	13	Rankle
7	Rwrist		

TABLE II. DATASET FOR TRAINING AND VALIDATION

Synthetic Data	Train (80%)	Validation (20%)
Fall	7690	1922
Non-Fall	42924	10731

TABLE III. TRAINING DETAILS

Model	Optimizer	Epoch	Bach Size
ST-GCN	AdaDelta	50	256

IV. Experimental Results

A. 2D Skeleton Evaluation

To evaluate the 2D skeleton, the mean Average Precision (mAP) of detected keypoints from synthetic fall videos was measured as shown in Fig. 5. The result of AlphaPose bottom-up approach was 72.3, utilizing the pretrained ResNet152 model on the COCO dataset. In contrast, skeleton data extracted by the proposed method is able to be considered 100 in terms of mAP since it is not estimation but actual measurement. Therefore, it can be expected that data trained on measured data will show better performance.

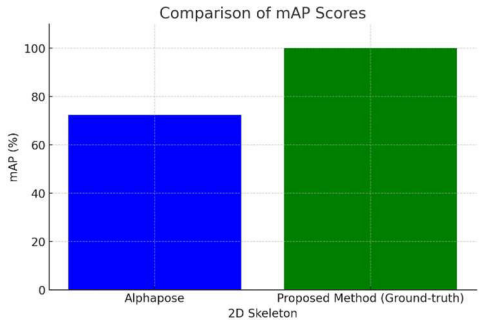


Fig.5. Skeleton mAP Comparison

B. Impact of Skeleton Quality on Fall Detection Performance

Fig. 6 presents the accuracy and F1-Scores of fall detection models. AlphaPose exhibited an accuracy of 79.86% and an F1-Score of 0.60, while using synthesized skeleton data through our proposed method resulted in an accuracy of 84.49% and an F1-Score of 0.73, demonstrating better performance as expected. The above results demonstrate an improvement in fall detection performance as the quality of the skeleton improves through our proposed method utilizing ground-truth skeleton data.

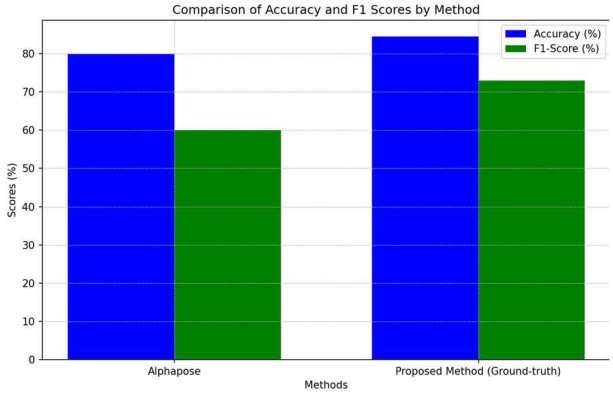


Fig.6. ST-GCN Results for Each Methods

V. Conclusion

The main focus of the proposed research was to demonstrate the relationship between skeleton quality and fall detection model performance, validating the effectiveness of measured (ground-truth) synthesized skeletons. The required skeleton dataset for experiments was constructed using a bottom-up pose estimation model and proposed method for synthesized videos extracted from Unreal Engine, with ST-GCN employed for fall detection classification. The experimental results showed a direct correlation between each pose estimation performance and the quality of skeleton data obtained from Unreal Engine and AlphaPose. This highlights the potential for superior performance of fall detection models with accurate skeleton data. Our proposed research methodology, however, did not account for the impact of different pose estimation algorithms on the quality of the skeleton. An analysis of how each pose estimation algorithm responds to synthetic data would be required, Future research aims to delve deeper into the analysis of various pose estimation models to further understand their impact on detection performance.

References

[1] Hwang, H., Jang, C., Park, G., Cho, J., & Kim, I. J. (2021). Eldersim: A synthetic data generation platform for human action recognition in eldercare applications. IEEE Access, 11, 9279-9294.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Xu, Y., Zhang, J., Zhang, Q., & Tao, D. (2022). Vitpose: Simple vision transformer baselines for human pose estimation. Advances in Neural Information Processing Systems, 35, 38571-38584.

[3] Fang, H. S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., ... & Lu, C. (2022). Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[4] Kreiss, S., Bertoni, L., & Alahi, A. (2021). Openpifpaf: Composite fields for semantic keypoint detection and spatio-temporal association. IEEE Transactions on Intelligent Transportation Systems, 23(8), 13498-13511. 37(1), 578-595.

[5] Osokin, D. (2018). Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. arXiv preprint arXiv:1811.12004.

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[9] Chhetri, S., Alsadoon, A., Al‐Dala'in, T., Prasad, P. W. C., Rashid, T. A., & Maag, A. (2021). Deep learning for vision‐based fall detection system: Enhanced optical dynamic flow. Computational Intelligence, 37(1), 578-595.

[10] Alanazi, T., & Muhammad, G. (2022). Human fall detection using 3D multi-stream convolutional neural networks with fusion. Diagnostics, 12(12), 3060

[11] Lin, C. B., Dong, Z., Kuan, W. K., & Huang, Y. F. (2020). A framework for fall detection based on OpenPose skeleton and LSTM/GRU models. Applied Sciences, 11(1), 329.

[12] Keskes, O., & Noumeir, R. (2021). Vision-based fall detection using st-gcn. IEEE Access, 9, 28224-28236.

[13] Anonymized. (Anonymized year). A Study on Synthetic Data Generation for Fall Detection . In 2024 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

turnitin Page 64 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Development of Open Source Big Data Technology using Project Management to addressing the complexity in ERP Implementation

1st Santo Fernandi Wijaya
Information Systems Department
Faculty of Informatics and Engineering
Universitas Multimedia Nusantara
Banten, Indonesia.
santo.fernandi@umn.ac.id

2nd Angelina Ervina Jeanette Egeten
Information Systems Department
Binus Online Learning
Bina Nusantara University
Jakarta, Indonesia.
angelina.egeten@binus.ac.id

3rd Jansen Wiratama
Information Systems Department
Faculty of Informatics and Engineering
Universitas Multimedia Nusantara
Banten, Indonesia
jansen.wiratama@umn.ac.id

***Abstract*—In the current digital era, implementing an Enterprise Resource Planning (ERP) system is one way to develop open-source technology for big data. So, ERP implementation is an absolute must for companies to improve performance. ERP implementation is an IT project that requires effective project management to achieve success in ERP implementation. Although previous studies have discussed the critical success factors in ERP project implementation, further research is still needed from a project management perspective in achieving successful ERP project implementation. This research methodology uses the Prisma method literature review of articles published in the last 5 years and found 31 articles as selected articles for data processing in finding important indicators in project management that influence ERP project implementation. The author validates the results of indicator mapping from a literature review with professional respondents and manages the data using the entropy method to determine indicator ranking. The results of this research can be used as a measuring tool to provide practical understanding for industry players, practitioners, academics, and consultants in overcoming project management challenges in implementing ERP projects. The results of this research are the development of a project management model for implementing ERP projects and developing open-source technology for big data.**

***Keywords*—Big Data, ERP Implementation, Open-source, Project Management.**

I. INTRODUCTION

For an organization, the use of information systems has become an absolute necessity which is used as a tool to support company operations and decision-making [1]. Companies need to carry out evaluations to understand the implementation of ERP systems to achieve success so that they can strategies carried out in implementing a successful and timely ERP system that can help company management in making strategic decisions [2]. It cannot be defined that implementing ERP is not just a project for the IT department, but needs to involve cross-related departments, such as marketing, financial accounting, human resources, production, and procurement. It can be said that the ERP project implementation process needs to consider adopting principles for managing effective project management. By understanding the principles of effective project management, company management can increase the success of implementing ERP projects promptly and minimize the risk of failure. The process of implementing ERP as an IT project is closely related to project management principles, such as carefully preparing each project management process effectively and efficiently so that the management project can be completed on time. This Research Question is to measure the relationship between ERP project implementation and project management modeling. Research questions answered in this study, namely as follows:

1. What important factors influence project management in implementing an ERP project?
2. How is a project management model developed for open-source big data technology of an ERP project?

II. BACKGROUND OF THEORETICAL

A. Challenge Project management in ERP project

An ERP system is an integrated information system that companies use to integrate and automate cross-departmental business processes to increase the company's operational efficiency [5]. The aim of implementing ERP is to improve the organization's business processes to become more efficient, effective, and productive. However, ERP implementation still provides challenges for organizations to achieve success in achieving go live on time. From previous research, there is still little research that examines from a project management perspective in implementing ERP as a project. Project Management is a series of planning and resource allocation processes to ensure the stages of the project management process run effectively, efficiently, and on time. Implementing an ERP project is a complex job because it integrates various modules, data, and business processes across departments and business units. For this reason, effective project management will help manage the complexity of ERP project implementation, because project management can detail project management steps, assign responsibilities, and develop realistic schedules [3]. Implementing an ERP project has significant challenges and risks, such as failure to integrate systems, changes to business processes that do not meet needs, and high resistance from users. By implementing effective project management, it can support the identification, evaluation, and management of failure risks, and minimize the impact on ERP implementation. The application of project management can support more effective planning and management of ERP project budgets, including the use of human and technical resources as needed so that resource use becomes more

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

efficient. Implementing project management can form a framework for evaluating ERP project implementation and taking corrective action to keep ERP project implementation on track. Implementing project management can help in planning, managing, and facilitating the changes needed to support the ERP project implementation running well. Implementing project management can increase collaboration in effective communication with stakeholders, thereby helping users adopt and understand how the ERP system works more effectively. The application of project management can support the preparation of a realistic ERP project implementation schedule, considering tasks and resource requirements, so that the implementation of each stage of the ERP implementation project can proceed according to the predetermined plan.

B. Project Management Framework

A project management framework represents a structured approach to planning, executing, monitoring, and controlling projects within an enterprise. The project management framework must be uniquely tailored to the specific needs and industry requirements of the company. In general, a project management framework will provide a visual representation of project management-related elements that assist project managers and team members in understanding and implementing effective project management practices. The project management framework describes the key competencies of the project. Project Managers must know, skills, and develop 10 areas of project management to achieve project success, namely: [4]

1. Project scope management involves defining and managing all the work necessary to complete the project successfully.
2. Project schedule management includes estimating the time required to complete work, developing an acceptable project schedule, and ensuring timely completion of the project.
3. Project cost management to prepare and manage the project budget.
4. Project quality management ensures that the project will meet the needs and objectives of project implementation
5. Project resource management is concerned with the effective utilization of resources people and physical resources involved with the project.
6. Project communications management involves generating, collecting, disseminating, and storing project information.
7. Project risk management includes identification, analysis, and response to risks associated with the project.
8. Project procurement management involves procuring goods and services for a project from outside the implementing organization.
9. Project stakeholder management includes the identification and analysis of stakeholder needs.
10. Project integration management is a comprehensive function that influences all other areas of knowledge.

C. Project Management is Not an IT Project.

Project Management is Not an IT Project. Project management is a scientific discipline with the scope of planning, organizing, and supervising the implementation of a project which includes the processes, methodology, and skills used to control the project so that it can achieve its goals. The project refers to a temporary set of activities to produce a unique product, service, or result within a specified time frame and budget. So, Project Management is not a project of the IT department, even though the activity process needs technological support. Project management is a set of practices and methodologies applied to ensure the successful completion of a project.ERP projects and project management are closely related, and the successful implementation of an ERP system depends heavily on effective project management practices [4]. Project Management is an integral part of the success of an ERP implementation project. In Project Management, the project managers play a central role in planning, implementing, and monitoring various aspects of ERP implementation to ensure that ERP implementation results are aligned with business objectives and provide benefits that meet stakeholder needs.

D. Previous Research

The authors carried out mapping using survey methods Prisma [6]. From survey previous research related to finding critical factors and gaps analysis of ERP projects with search engines: Semantics scholar, Scopus, Springer, Emerald Insight, and IEEE Xplore, with a search formulation using the keywords: "challenge project ERP, project management". Based on a review of previous research, the authors created inclusion criteria: Discussion of research that focuses more on ERP projects in the ERP implementation and project management stages, published in international conferences and journals published from the period 2018 to 2022. As for the exclusion criteria: research does not discuss opinions, suggestions, discussions, and presentation papers. Based on the mapping results from the literature study, considering records exclusion and records inclusion, of the 727 articles found through database sources search, 31 articles were articles ready obtained for further analysis to obtain indicators related to project management and ERP projects. The literature study mapping work process can be seen in Figure 1.

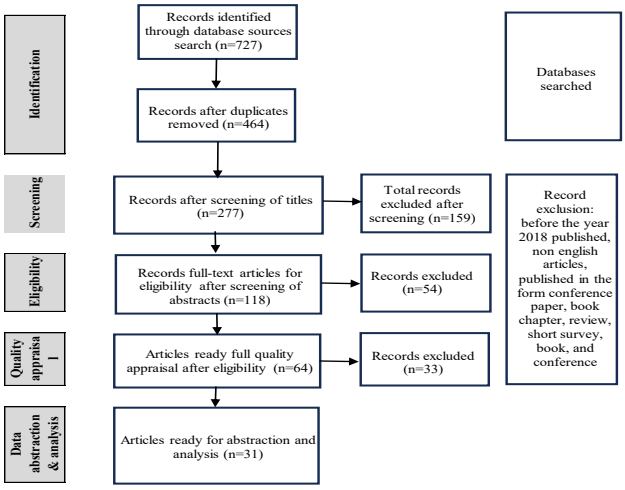


Fig. 1. Flow of Literature Review

Based on the findings from 31 articles ready for analysis, after carrying out the mapping process, 18 critical indicators

related to ERP projects and project management were found. After that, based on these indicators, authors create definitions based on the selected articles.

Based on the results of the literature study indicator mapping related project management effectively, the authors defines these indicators as follows:

Implementation strategy in project management can provide careful planning, stakeholder involvement, identify risk management, and flexibility in adapting strategies to changing business needs [3], [7], [8], [9], [10]. Process Change in project management can manage process changes in ERP implementation including business process analysis, and active involvement of stakeholders [3],[10],[11],[12]. Risk management process in project management can identify risks to changes in company policy so that it can minimize the level of user resistance to changes in business processes [5]. Project management method in project management can minimize the complexity of implementing an ERP project, including system integration, business process changes, and data conversion [3], [8],[10],[13],[14]. Integrated business process in project management creates integrated business processes such as identifying inter-process relationships, modeling comprehensive processes, and monitoring the success of integration [5], [10], [15],[16],[17],[18]. Sharing Knowledge in project management can facilitate knowledge sharing in ERP projects such as the creation of a collaborative culture, and the establishment of mechanisms for sharing information [19],[20]. Human Resources can involvement of end users who are actively involved in the ERP implementation process can minimize the level of user resistance [21]. User Training can adequate user training for end users can provide an understanding of how to use the ERP system, thereby increasing success in implementing ERP project [3], [10]. Top management support is active support and involvement from executive management, so that ERP project implementation needs can be met [3],[10],[19],[22],[23],[24]. Vendor /consultant selection in selecting the right and experienced vendor according to the company's characteristics and needs can minimize failure in implementing an ERP project [3], [10]. Change management in project management in carrying out change management by business needs, with a comprehensive understanding of business needs for the adoption of selected ERP systems that can support the implementation of ERP projects running well [3], [9], [22], [25], [26]. Communication effective in project management in improving effective communication in good control over the progress of ERP implementation projects regularly can support the implementation of ERP projects well [3],[25]. Readiness measurement in careful planning involves the project team in determining project goals, understanding business needs, and planning implementation steps in detail [27],[33]. Project budget plan in project management can plan the ERP project budget and allocate adequate budget for various project needs [3], [28],[29]. Operating systems in project management can identify operational system needs, testing, and integration with existing systems [30]. Data integrity in project management can maintain data integrity in ERP projects such as accurate data mapping, consistent testing of data [3],[16]. IT Infrastructure in project management can manage IT infrastructure in ERP projects such as scalable and flexible infrastructure planning and design, testing and optimizing infrastructure performance, and infrastructure availability [3]. Technology support in project management can provide technological support in ERP projects such as selecting technology that suits business needs [30], [31].

Summary of the results of the literature study indicator mapping, it can be said that aspects of Project Management ranging from project planning to change management are very important to ensure the ERP project runs. Project Management and Successful ERP Project Implementation Implementing effective project management can support determining criteria for achieving success in an ERP implementation project, namely by evaluating ERP project objectives, user satisfaction, and company performance according to project management principles. Based on the definition of the factors from the mapping results, the authors created a questionnaire statement to be tested on the respondents. The questionnaire statement can be seen in Table I.

TABLE I. LIST OF QUESTIONNAIRE STATEMENT

No	Definitions	Questionnaire statement
1	Implementation strategy	How important Implementation strategy for achieving implementation ERP success?
2	Process Change	How important process changes in an ERP project become a process to optimize organizational operations?
3	Risk Management Process	How important risk management process become a process that can help organizations navigate the uncertainties and challenges associated with ERP projects?
4	Project Management Method	How important Project management method for guiding the project team, managing risks, and ensuring ERP implementation success?
5	Integrated Business Process	How important Integrated Business Processes can create systems that improve operational efficiency, decision making and organizational performance?
6	Sharing of Knowledge	How important Effective knowledge sharing for building a collective understanding of the ERP project?
7	Human Resource Management	How important does HRM play a role in properly preparing employee needs in supporting the ERP project?
8	User Training	How important User training becomes critical for maximizing the benefits of ERP projects?
9	Top Management Support	How important Strong top management support in determining factor in the success of an ERP project?
10	Vendor/Consultant selection	How important the right vendor or consultant Selection for enhancing the ERP project success?
11	Change Management	How important is Change Management in preparing, supporting, and helping organizations make organizational changes?
12	Communication Effective	How important Communication effective for achieving the goals of the ERP project?
13	Readiness Measurement	How important is Measuring Organizational Readiness to ensure aspects such as the availability of resources to achieve project goals?
14	Project Budget Plan	How important is the Project Budget Plan for implementing effectively with the availability of adequate resources?
15	Operating Systems	How important Operating system for an ERP project becomes a strategic decision for the ERP systems requirement?
16	Data Integrity	How important has Data integrity become for ensure that the data in the ERP system is accurate and complete?
17	IT Infrastucture	How important is IT infrastructure for the ERP Project's success?
18	Technology Support	How important Technology support for reliable functioning of the ERP system?

III. METHODOLOGY

A. Research Method

This research method uses quantitative methods [33]. By using a questionnaire with the Google Form tool to collect data from respondents. Using a purposive sampling technique based on the selected respondent population makes it easier to collect respondent data and can provide information that is right on target for the research object [34]. The research results of data management use Entropy as a data processing tool to measure the level of uncertainty or randomness in a data set. The Entropy Method serves to analyze data and improve the quality of model predictions, which can be used to rank critical factors in ERP project implementation. [36]. The respondents in this research were employees and professional workers in project management and ERP. The authors explain the Likert scale levels used and the characteristics of respondents in the results section of this research. The data survey was collected by google forms that has been sent to respondents by online platform using Semantic differential scale, and the open data research was available on the link https://doi.org/10.5281/zenodo.13124648.

B. Research Model

Based on the definition of components of project management from previous studies, the authors made a summary statement related to project management for ERP projects, namely that Project risk management is a dynamic and ongoing process that requires collaboration, analysis, and adaptability. Project Scope Management is essential for project success for the project will deliver. Project Human Resource Management requires a strategic approach to acquiring, developing, and managing the project team. Project stakeholder management is essential for navigating the complexities of project environments. Project Procurement Management is integral to the project's success in meeting project objectives. Project Quality Management is an essential aspect of project management and contributes to project stakeholders' success. Project cost management involves a continuous cycle of planning, monitoring, and controlling to ensure the project's approved budget. And Project Integration Management is fundamental to aligning with objectives and contributing to project success.

Based on the definition of project management components, the author designed designed the development of a project management model that can be used in implementing ERP as a project. The design of the project management model for ERP projects implementation can be seen in Table II.

TABLE II. PROJECT MANAGEMENT FOR ERP PROJECT MODEL

ERP Project		Project Management
Dimension	**Indicators**	
Process Business	Implementation Strategy	Project Risk Management
	Process Change	
	Risk Management Process	
	Project Management Method	Project Scope Management
	Integrated Business Process	
	Sharing of Knowledge	
People	Human Resource Management	Project Human Resource Management
	User Training	
	Top Management Support	Project Stakeholder Management
	Vendor/Consultant Selection	Project Procurement Management
Organizational	Change Management	Project Quality Management
	Communication Effective	
	Readiness Measurement	
	Project Budget Plan	Project Cost Management
Technology	Operating Systems	Project Integration management
	Data Integrity	
	IT Infrastructure	
	Technology Support	

C. Entropy Method

Based on the design of the project management model for ERP projects implementation in Table 2, Authors processed the data using the entropy method. The Entropy method is a method for determining the ranking of criteria that can be used for decision-making [35]. This research uses the entropy method to determine weighting criteria to express the probability distribution spread across a questionnaire statement. The stages of weighting criteria using the entropy method are as follows:

1. All respondents gave a value indicating the importance of a particular criterion determined in the questionnaire statement.
2. Subtract each of these numbers by the ideal value. The result can be expressed in Xij.
3. The Xij value is obtained from the Pij matrix as follows.
$$P_{ij} = \frac{Xij}{\sum_{i=1}^{m} X_{ij}}, \forall i, j.$$
m = number of respondents
4. Calculate the entropy value for each criterion with the following formula:
$$Ej = -k \sum_{i=1}^{m} Pij \ln Pij, \forall j,$$
Where = $K = \frac{1}{lnm}$
Then calculate the spread of each criterion with the following formula:
$$dj = 1 - Ej, \forall j.$$
5. It is assumed that the total weight = 1, so to get the weight for each criterion, the dispersion value is normalized, with the following formula:
$$Wj = \frac{dj}{\sum_{j=1}^{n} dj, \forall j.}$$
n = number of criteria

IV. RESULT AND ANALYSIS

A. Result

This research uses a 1-6 level likert scale to measure the perceptions of respondents [36]. Scale 1 is very unimportant, scale 2 is not important, scale 3 is less important, scale 4 is quite important, scale 5 is important, and scale 6 is very important. The sampling process involves distributing questionnaires to respondents who understand the scope of

ERP and IT. In this research, the characteristics of the respondents consist of 94 respondents provided feedback on the questionnaire statements distributed via Google Forms. Respondent positions were dominated by staff level with 80 respondents (67%), and manager level with 9 respondents (28%). and director level as many as 5 respondents (5%). Regarding the experience of working using the ERP system the respondents, most respondents had 3-4 years of experience, 60 respondents (58%), having 5-6 years of experience with 33 respondents (32%), and had more than 7 years experience with 11 respondents (11%). In terms of the educational level of the respondents, most respondents had bachelor's degrees as many as 53 respondents (61%), master's degrees as many as 38 respondents (37%), and doctor's degrees as many as 3 respondents (3%).

B. Analysis

Based on the results of Entropy data processing, it shows that the 3 indicators that have significant scores are the Project Management Model ranking score of 0.976; Integrated Business Process score of 0.938; The Readiness Measure has a score of 0.900. where these indicators are included in the ERP project process dimensions. Regarding the project management component, the project management model and Integrated Business Process indicators are included in the project scope management, and Readiness Measures are included in the project quality management component. It can be said that to ensure an ERP project can achieve success and be on time, it is necessary to seriously consider business process re-engineering. The results of research for ERP projects can be seen in Table III.

TABLE III. RESULT RESEARCH

No	Indicators	Entropi Index
1	Project Management Method	0.976
2	Integrated Business Process	0.938
3	Readiness Measurement	0.900
4	Implementation Strategy	0.899
5	Top Management Support	0.898
6	Sharing of Knowledge	0.891
7	Risk Management Process	0.880
8	User Training	0.876
9	Vendor/Consultant Selection	0.876
10	Communication Effective	0.872
11	Operating Systems	0.871
12	Data Integrity	0.869
13	Human Resource management	0.867
14	Process Change	0.863
15	Technology Support	0.839
16	IT Infrastructure	0.799
17	Project Budget Plan	0.761
18	Change Management	0.751

Based on the results of this research, to answer the first research question it can be said that the 5 important factors that influence project management in ERP project implementation are: project management method, integrated business process, readiness measurement, implementation strategy, and top management support. Those factors have been through data analysis for later generating the open source with using one of big data technology like an Apache hadoop to carry out a large data processing, storage, and distribution in computer cluster because the ERP project implementation needs a big data processing to running all those, and as the same time this is a strategy on how to implement it.

To answer the second research question, where development of a project management model for generating open source using big data technology of an ERP project is to increase the success of ERP project implementation, the development of project management modeling can be a strong foundation in implementing a company's ERP project, namely by paying attention to components in Project Management such as Project Risk Management, Project Coverage. Management, Project Human Resources Management, Project Stakeholder Management, Project Procurement Management, Project Quality Management, Project Cost Management, and Project Integration Management.

V. CONCLUSION

The results of this research can identify that implementing an ERP project in a company is closely related to effective Project Management. This is the basis for comprehensively understanding the company's feasibility in deciding to accept the change process from a human, business process, organizational and technological perspective in developing open-source using big data technology. The challenge in achieving successful ERP project implementation is managing project management effectively which includes Project Risk Management, Project Scope Management, Project Human Resources Management, Project Stakeholder Management, Project Procurement Management, Project Quality Management, Project Cost Management, and Project Integration Management for generating an open source using big data technology to implement the ERP project at the company. The authors realizes that this research still has limitations, including limited industry coverage and has not yet produced an ERP application by proving the Development of Open-Source using big data technology with Project Management to address the complexity in ERP Implementation. This is an opportunity for other researchers to develop this research by expanding the scope of the industry by adopting the use of the latest technology, such as the use of artificial intelligence and ERP intelligence to increase user confidence in the effective Development of Open-Source using Big Data Technology like a Apache hadoop with Project Management.

ACKNOWLEDGMENTS

This research was supported by the Department of Research and Community Services, Universitas Multimedia Nusantara. We thank the Software Engineering Laboratory of Universitas Multimedia Nusantara, part of the Information Systems Department, for providing insight and expertise that greatly assisted the research.

AUTHOR CONTRIBUTION

Wijaya S.F presented the Introduction section, criticized, and mapped the Literature Review, determined the Research Methodology, analyzed the Research Results, summarized the research results in the Conclusion section, and revised the final examination for all necessary sub-sections. Egeten A.E.J is actively involved in analyzing and looking for indicators from study literature reviews, making

questionnaire statements and distributing them to respondents, processing data for validity and reliability tests, conducting hypothesis tests and explaining in detail the results of data processing. Wiratama J is participating in literature review, conclusion, and explained the limitations of the research and further research. All authors reviewed and approved the final version of the manuscript.

REFERENCES

[1] J. Kurniawan, and W. Wella, "Information Technology Governance Capability at PT XYZ using COBIT 2019," *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, 14(2), 58-65. 2023. DOI: https://doi.org/10.31937/si.v14i2.3223.

[2] W.M. Nurrohmah, and J. Wiratama, "Enterprise Resource Planning (ERP) SAP Business One Evaluation and Improvement Recommendation using Customized Odoo," *Ultima InfoSys: Jurnal Ilmu Sistem Informasi,* 13(2), 77-84. 2022. DOI: https://doi.org/10.31937/si.v13i2.2802.

[3] Q. Huang, M.M Rahim, S. Foster, and M. Anwar, "Critical Success Factors Affecting Implementation of Cloud ERP Systems: A Systematic Literature Review with Future Research Possibilities," *Hawaii International Conference on System Sciences*. 2021.

[4] K. Schwalbe, "*Information technology project management (9th edition),"* Cengage. 2019.

[5] T.S. Kiran, and A.V. Reddy, "Critical Success Factors of ERP Implementations in SMEs," *Journal of Project Management,* pp. 267-280. 2019. DOI: 10.5267/j.jpm.2019.6.001

[6] A. Liberati, D.G. Altman, J. Tetzlaff, C. Mulrow, P.C. Gøtzsche, J.P. Ioannidis, and D. Moher, "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration," *Annals of Internal Medicine*, 151(4), W-65. 2019. https://doi.org/10.7326/0003-4819-151-4-200908180-00136.

[7] A. Hanindito, T. Raharjo, B. Hardian, and A. Suhanto, "Agile Readiness Measurement in Organizations using Agile Adoption Framework: A Case Study on Indonesian Automotive Company," *International Conference on Informatics, Multimedia, Cyber and Information Systems (ICIMCIS* (pp. 162-168). IEEE. 2021.

[8] Alavi, S. Peivandzani, and S. Mirmohammadsadeghi, "Risk Assessment and Prioritization of ERP Implementation Based on BSC. *Journal of Human, Earth, and Future*. 2021.

[9] S. Madanian, M. Subasinghage, and S.C. Tachiona, "Critical Success Factors of Agile ERP Development and Implementation Projects: A Systematic Literature Review," *Pacific Asia Conference on Information Systems*. 2021.

[10] S.F. Wijaya. H. Prabowo, and R.R Kosala, "Agile Methods for ERP Implementation: A Systematic Literature Review," *International Conference on Information Management and Technology (ICIMTech)* (pp. 571-576). IEEE. 2018.

[11] O. Alaskari and M.M.R. Ahmad, "Framework for Selection of ERP System: Case Study," *Journal International Conference on Flexible Automation and Intelligent Manufacturing*, vol. 38, pp. 24-28, 2019. https://doi.org/10.1016/j.promfg.2020.01.009.

[12] J. Mc Donagh, and D. Saxena, "Evaluating ERP Implementation: The Case for a Lifecycle-based Interpretive Approach," *The Electronic Journal of Information Systems Evaluation*, pp. 29-35. 2019. https://academicpublishing.org/index.php/ejise/article/view/126.

[13] K. Kumar, and G.D.M., "An Overview of Production Planning and Control in the Make-to-Order Industry," *Research Journal of Engineering Technology and Medical Sciences*, pp. 320-325. 2022. DOI: http://www.rjetm.in/RJETM/Vol05_Issue03.

[14] T. Raharjo, and B. Purwandari, "Agile project management challenges and mapping solutions: A systematic literature review," In *Proceedings of the 3rd International Conference on Software Engineering and Information Management* (pp. 123-129). 2020.

[15]. A.N., and S. Mohmed, "Proposed Risk Diagnosing Methodology for ERP Implementation Project in SMEs," *American Journal of Business and Operations Research.* 2021.

[16] M. Ellison, R. Calinescu, and R.F. Paige, "Evaluating Cloud Database Migration Options Using Workload Models," *Journal of Cloud Computing*. 2018. https://doi.org/10.1186/s13677-018-0108-5.

[17] D.G. Schniederjans, "Business Process Innovation on Quality and Supply Chains," *Business Process Management Journal,* vol. 24, no. 3, pp. 1-17. 2018. https://doi.org/10.1108/BPMJ-04-2016-0088.

[18] O. Nafisa, and S. Abd-El-Kader, "A Software Requirement Engineering Framework to Enhance Critical Success Factors for ERP Implementation," *International Journal of Computer Applications*, vol. 180, no. 10. 2018. DOI: 10.5120/ijca2018916170.

[19] Jiwasiddi and Mondong, "Analyzing ERP Implementation Success Factors," *Pertanika Journal*, pp. 139-146. 2018. DOI: http://www.pertanika.upm.edu.my/.

[20] Uddin, M.S. Alam, and Z.K.T.U Al-Mamun, "A Study of the Adoption and Implementation of Enterprise Resource Planning (ERP): Identification of Moderators and Mediators," *J. Open Innov. Technol. Mark. Complex*, vol. 6, no. 1. 2020. https://doi.org/10.3390/joitmc6010002.

[21] L.W. Q Ping, and W. Liu, "Enterprise Human Resource Management Platform Based on FPGA and Data Mining," *Microprocessors and Microsystems Journal*, vol. 80, 2021. https://doi.org/10.1016/j.micpro.2020.103330.

[22] M. Ahmed Kabbary, and N.A. Jawawi, "Preventing Enterprise Resource Planning Failure Through an Enhanced Approach to Solve Ineffective Communication," *International Journal of Innovative Computing*. 2021.

[23] H. Bruverud, "ERP system implementation: How top managers' Involvement in a change project matters," 2019.

[24] M. Lakdinu, M.M. De Silva, K. Balachandra, and K.P. Frank Perera, "Leadership in Change Management Decisions for Successful ERP Implementation - A System Dynamics Perspective," *Moratuwa Engineering Research Conference* (MERCon), 1-6. IEEE 2022.

[25] R.J. Sweis, R. Abuhussein, and D. Jandall, "Factors Affecting ERP Projects from a Project Management Perspective," *International Journal of Business Information Systems*, pp. 281-296. 2018. https://doi.org/10.1504/IJBIS.2018.095564.

[26] Z. Kazemi, and M. Tavana, "A Fuzzy Group Decision Making Approach for ERP System Selection: An Empirical Study," *Journal of Enterprise Information Management,* vol. 31, no. 2, pp. 312-330, 2018. https://doi.org/10.1504/IJBIS.2014.060836.

[27] Z. Fiona, J.C. Shereen, S. Juniarty, and J. Gunadi, "Implementation of ERP System for Inventory Management at PT XYZ," *Mirai Management Journal*, vol. 7, no. 3, pp. 1-10. 2022. DOI: https://doi.org/10.37531/mirai.v7i3.2962.

[28] J. Dongmin and S. Moonsoo, "Development of Distributed MRP System for Production Planning and Operation in Korean OEM/ODM Cosmetics Manufacturing Company," *Journal of the Society of Korea Industrial and Systems Engineering*, pp. 133-141. 2020. DOI: https://db.koreascholar.com/Article/Detail/404272.

[29] M.N. Yakubu, and S.I. Dasuki, "Assessing Elearning Systems Success in Nigeria: An Application of The Delone and Mclean Information Systems Success Model," *Journal of Information Technology Education*: *Research*, no. 17, pp. 184-198. 2018. DOI: https://doi.org/10.28945/4077.

[30] R. Dastres, and M. Soori, "Advances in Web-Based Decision Support Systems," *International Journal of Engineering and Future Technology.* 2022. DOI: https://hal.science/hal-03367778/.

[31] M.S.N. Kabir, M.N. Reza, M. Chowdhury, M. Ali, Samsuzzaman, M.R. Ali, and S.O. Chung, "Technological Trends and Engineering Issues on Vertical Farms: A Review," *Horticulture*, 9(11), 1229. 2023.

[32] M. Hartono, and H. Susanto, "A Corporate Sustainability Maturity Model for Readiness Assessment: a Three-Step Development Strategy," *Department of Industrial Engineering, University of Surabaya Journal*, pp. 1-25. 2020. DOI: https://doi.org/10.1108/IJPPM-10-2019-0481.

[33] I. Hermawan, "Metodologi Penelitian Pendidikan (Kualitatif, Kuantitatif, dan Mixed Method)," Hidayatul Quran, 2019

[34] C. Andrade, "The Inconvenient Truth About Convenience and Purposive Samples," Indian J Psychol Med., vol. 43, no. 1, pp. 86-88, 2021. DOI: 10.1177/0253717620977000.

[35] M.B. Khalilzadeh, H.R. Youshanlouei, M.M. Mood, "Identifying and ranking the effective factors on selecting Enterprise Resource Planning (ERP) system using the combined Delphi and Shannon Entropy approach," *Procedia - Social and Behavioral Sciences*, 41, 513–520, 2012.

[36] S.H.F.I. Edi and H. Dasril, "Analysis of Satisfaction Level Using Likert Scale on Speedy Services that Migrated to Indihome," *Electro Engineering Journals*, vol. 1, no. 1, 2019.

turnitin Page 70 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

An Improvement on Exploration Step of Whale Optimization Algorithm with Levy Distribution for Classification Problems

Sakkayaphop Pravesjit, Krittika Kantawong, Natdanai Kamkhad, Saksit Sabaiporn, Jantawan Monchanuan
School of Information and Communication Technology University of Phayao
Phayao, Thailand
sakkayaphop.pr@up.ac.th, krittika.ka@up.ac.th, nattapong.ka@up.ac.th, taesaksit09@gmail.com, jantawann.00@gmail.com

Duangjai Jitkongchuen
Big Data Instutite
Bangkok, Thailand
duangjai.ji@bdi.or.th

Arit Thammano
Facluty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Thailand
arit@it.kmitl.ac.th

Panchit Longpradit
Faculty of Social Sciences and Humanities
Mahidol University
Nakhon Pathom, Thailand
panchit.lon@mahidol.ac.th

Abstract— The proposed system represents an enhanced movement in search of food of Whale Optimization Algorithm (WOA), based on Levy distribution for image classification of grape leaf disease. In the preprocessing, the proposed system uses convolution kernels to transform images into input data within the range of (0,1). Thereafter, the Levy distribution is incorporated into the WOA model as an exploration search mechanism. A grape leaf dataset from the Plant Village project (www.plantvillage.org), consisting of 4062 labeled images with dimensions of 256 by 256 pixels and divided into four classes –healthy, Black Rot, Black Measles, and Isariopsis leaf spot –is used to evaluate the performance of the proposed system. Experimental results show that the proposed system is better than Visual Geometry Group (VGG16), Gray Level Co-occurrence Matrix (GLCM) with SVM, Low contrast haze reduction-neighborhood component analysis with SVM, and whale optimization algorithm (WOA).

Keywords—whale optimization algorithm, levy distribution, convolution kernel, grape leaf disease

I. INTRODUCTION

Plant diseases are a main problem in agricultural production, which has negative effects on the quality and yield of plants [1]. Traditional plant disease detection is hindered by the expertise of farmers [2], who are often constrained by their professional knowledge and experience. As a result, it is difficult to quickly and accurately identify the disease. From the above statement, grape plants are also affected by diseases such as powdery mildew, brown blotch, and anthracnose, which can significantly affect the yield and quality of grapes. Inspection of grape plant diseases relies on farmers' own observation of symptoms on plant leaves or taking pictures of grape plant leaves and sending them to experts for diagnosis [3]. This is because diseased grape leaves often exhibit visible spots.

Currently, artificial intelligence technology has been widely applied in plant disease classification in agriculture, for example computer vision, evolution algorithm, deep learning algorithm, etc. Nitesh et al . [4] presented the transformation method of the grape leaf disease images into HSV color space for feature extraction. This was then followed by detection of the disease using gray-level occurrence matrix-based features and support vector machine. Nasiri and Nejand [5] presented the method for grape leaf disease detection, consisting of 2 steps: segmentation step using Gray-Level Co-occurrence Matrix (GLCM), Gray Level Run Length Matrix (GLRLM), and classification step using Local Binary Pattern (LBP) based features. Javidan et al. [6] presented grape leaf disease classification, whereby the image processing involves the use of k-means clustering to segment the region of interest for grape leaf disease. Subsequently, RGB, HSV and LAB colors were utilized for feature extraction. Finally, SVM was employed for classifying the type of disease. Ashokkumar et al. [7] demonstrated the proposed method for grape leaf disease detection using a faster region-based convolutional neural network. Their main idea combines attention-based multilayer convolutional feature creation, object identification, and categorization.

This paper presents an improvement on WOA for classification of grape leaf disease. The performance of the proposed algorithm was compared with VGG16, GLCM with SVM, GoogleNet&ResNet50, and AlexNet.

Following this introduction, Section 2 presents a dataset description. In Section 3, whale optimization is described. Section 4 presents the proposed algorithm. In Section 5, the experimental results are presented and discussed. Finally, Section 6 concludes the study.

II. DATASET DESCRIPTION

This study utilizes a publicly available dataset from the Plant Village project (www.plantvillage.org). The grape leaf dataset was chosen, comprising 4062 labeled images with dimensions of 256 by 256 pixels, then separated into two subsets, healthy and diseased, and further categorized into three distinct diseases: Black Rot, Black Measles, and Isariopsis leaf spot. Table 1 presents the sample images for each class.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

turnitin Page 71 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

TABLE I. SAMPLE IMAGES FOR EACH CLASS FROM THE PLANT VILLAGE DATASET.

Input class	Image samples
Healthy	
Black Rot	
Black Measles	
Isariopsis leaf spot	

III. WHALE OPTIMIZATION ALGORITHM

The Whale Optimization Algorithm (WOA) was presented by Mirjalili and Levis [8] in 2016. Fig. 1 exhibits a flowchart diagram of WOA. The process of WOA includes search and attack their prey. First, the exploitation phase of the algorithm involves attacking the prey. This phase consists of updating the solution's location through the process of the whale population surrounding the prey and executing the spiral bubble net attack. Spiral location updates and searches for victims. Second, exploration phase involves whales randomly searching for food. The processes are explained in detail as follows:

(i) Encircling Prey: in the case where p < 0.5 and |A| < 1, it explains whales surround the prey after identifying its location. The whale with the best fitness value is considered to be at the prey location, while the other whales surround the prey and update its new position using the following equation:

$$x_{t+1} = x_{Gbest} - A \times D \quad (1)$$

$$\mathrm{A} = a \times (2 \times rand - 1) \quad (2)$$

$$C = 2 \times rand \quad (3)$$

$$D = |C \times x_{Gbest} - x_t| \quad (4)$$

$$a = 2 \times \left(1 - \frac{t}{t_{max}}\right) \quad (5)$$

turnitin Page 72 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

$$l = \left[\left(-1 + \left(\frac{-t}{t_{max}}\right)\right) - 1\right] \times rand + 1 \quad (6)$$

where x_t represents the current location of the whale and x_{t+1} denotes the whale's new location. x_{Gbest} signifies the location of whale closest to the prey. a is a linear convergence factor and its value decreasing from 2 to 0 from the increasing number of iterations and t is the number of iterations in this present time, t_{max} is the maximum of the iterations. $rand$ is a random value in the range (0,1).

(ii) Search for Prey: If p < 0.5 and |A| ≥ 1, in addition to the predation behavior by bubble net, whales can also engage in random food search. The exploration stage of the algorithm involves the process of searching for food. Individuals whale randomly search for food according to each other's positions, and the mathematical model can be expressed as follows:

$$D = |C \times x_{rand}(t) - x_t| \quad (7)$$

$$x_{t+1} = x_{rand}(t) - A \times D \quad (8)$$

where $x_{rand}(t)$ represents the location of random individual whale.

(iii) Spiral Updating Position: If p ≥ 0.5, during the WOA exploitation process, the humpback whale updates its position around the prey using its unique blister-like spiral behaviors. The spiral update position is expressed by the following equation:

$$D' = |x_{Gbest} - x_t| \quad (9)$$

$$x_{t+1} = D' \times e^{bl} \times \cos(2\pi l) + x_{Gbest} \quad (10)$$

where D' is the distance between the i^{th} candidate solution and the best solution in the current generation, b is a constant and $l \in [-1,1]$.

The WOA can efficiently solve the optimization problem. This is especially true for low-dimensional functions. However, it still has some drawbacks in dealing with the height dimension function. The performance of the algorithm decreases significantly and is easily trapped in the local optima because it tends to initially converge very quickly.

turnitin Page 72 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

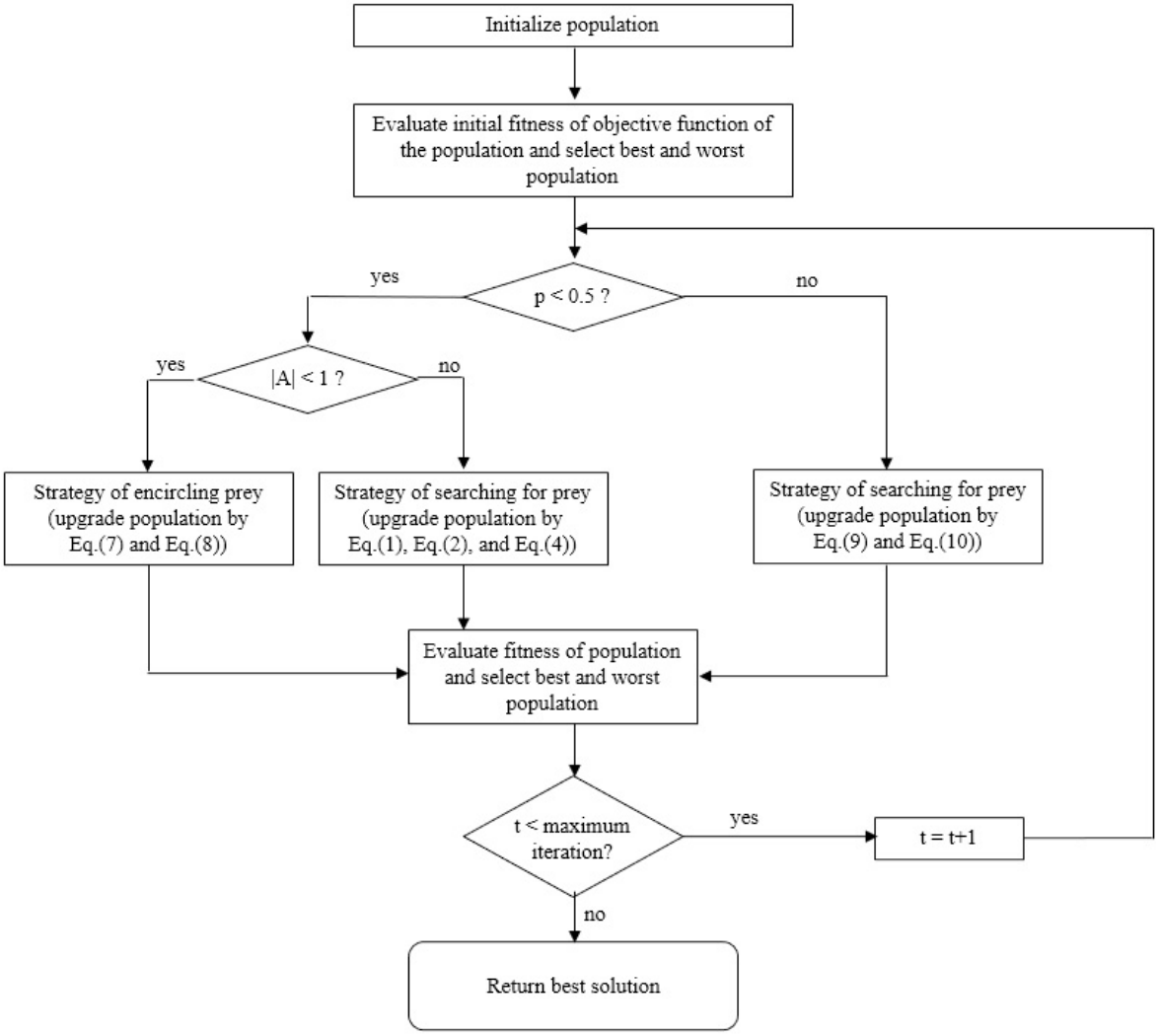


Fig. 1. The flowchart of the Whale Optimization Algorithm.

IV. THE PROPOSED ALGORITHM

This paper is aimed at improving the exploration of position updating of Whale Optimization Algorithm for classification problems. The flowchart of the proposed algorithm is demonstrated in Fig 2. After initial populations and evaluation of their fitness values, the Levy distribution is applied to improve procedures of exploration of position updating. The processes are explained in detail as follows:

A. Lavy distribution

In this study, the process used random numbers with Levy Flight (LF), which consists of two phases. The first phase involves selecting the random direction, and the second phase takes into consideration the generation of steps that follow the chosen Levy Distribution (LD) [9]. The selection of random direction is drawn from a uniform distribution. The step lengths can be calculated by:

$$LD = \frac{u}{|v|^{1/\beta}} \tag{11}$$

where u and v are drawn from normal distributions. This implies that:

$$u = N(0, \delta_u^2, n, m) \tag{12}$$

$$\delta_u = \left\{ \frac{\tau(1+\beta)\sin\left(\pi\beta/2\right)}{\tau\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{13}$$

$$\beta \text{ is constant value with } 0 < \beta < 2 \tag{14}$$

$$v = N(0,1,n,m) \tag{15}$$

Following this, the population of LD and the whale population under consideration was selected from the Eq. (16).

$$x_{t+1} = max_{fitness} \begin{cases} D' \times e^{bl} \times \cos(2\pi l) + x_{Gbest} \\ LD \times e^{bl} \times \cos(2\pi l) + x_{Gbest} \end{cases} \tag{16}$$

turnitin Page 74 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

From Eq. (16), the process is to select the position of whale with the highest fitness value to use in this exploration step.

TABLE II. THE PARAMETERS USED IN THE EXPERIMENTS

Parameters	Values
Population size, NB	25
Dimension, (ND)	100
Max iteration	300
# of Data train	4062 – (# of Data test)
# of Data test	10% of each class data sets.

TABLE III. THE PARAMETERS USED IN THE LEVY DISTRIBUTION

Parameters	Values
Population size, NB	25
Dimension, (m)	100
Max iteration	300
# of data class (n)	4

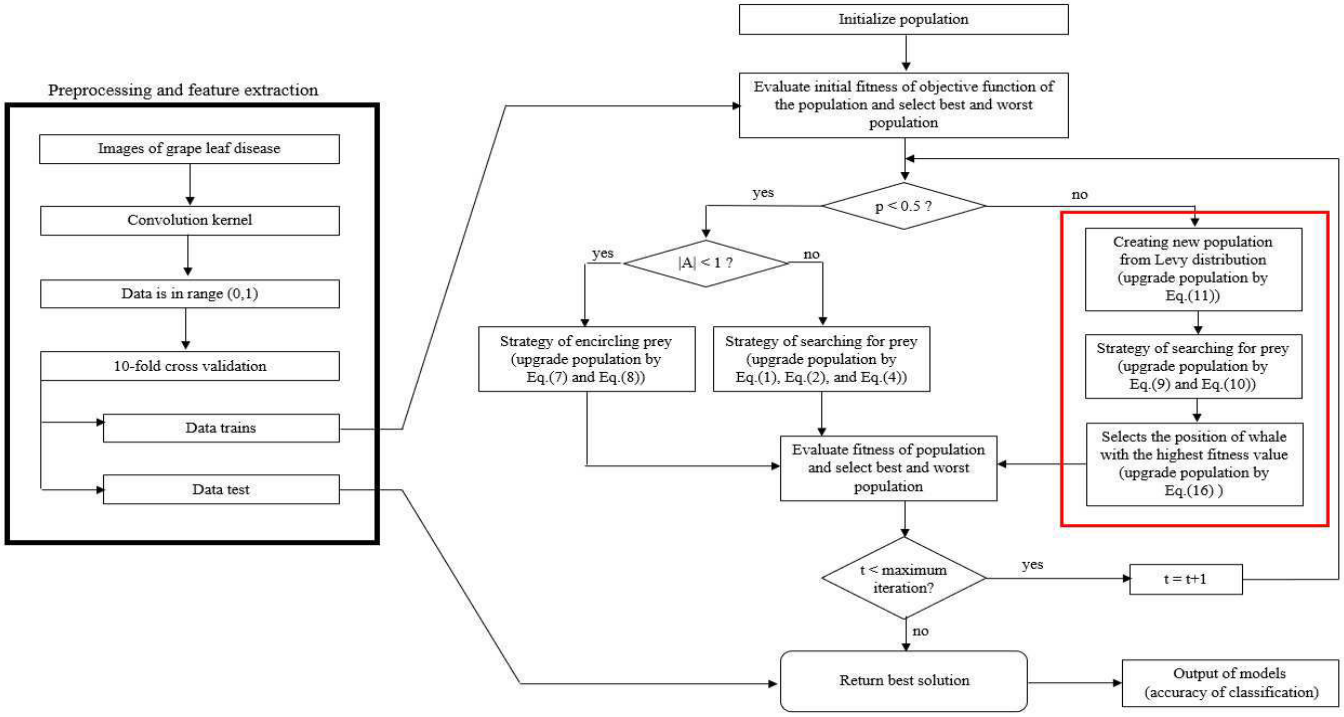


Fig. 2. The flowchart of the proposed algorithm.

V. THE EXPERIMENTAL RESULTS

From Table 1, the grape leaf dataset includes 4062 labeled images with dimensions of 256 by 256 pixels. The grape leaf dataset is separated into four classes, consisting of healthy, Black Rot, Black Measles, and Isariopsis leaf spot.

The computational results shown in Table 4 present a comparison of the proposed algorithm with VGG16 from Yohan Rayhan et al. [10], GLCM with SVM from Sajjad Nasari et al. [11], Low contrast haze reduction-neighborhood component analysis with SVM by Alishba Adeel et al. [12], VGG16 by Arie Hasan et al. [13], and original WOA from Mirjalili and Levis [8]. The finding indicates that the proposed algorithm has offered better results than the other five compared algorithms.

TABLE IV. COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE PROPOSED WORK WITH THAT OF OTHER EXISTING WORKS.

Sl.no	Source	Methodology	Accuracy (in %)
1	Yohan Rayhan et al. [10]	VGG16	87.50
2	Sajjad Nasari et al. [11]	GLCM with SVM	89.93
3	Alishba Adeel et al. [12]	Low contrast haze reduction-neighborhood component analysis with SVM	91.34
4	Arie Hasan et al. [13]	VGG16	95.00
5	Mirjalili and Levis [8]	Original WOA	86.81
6	Proposed algorithm	LD-WOA	98.73

VI. CONCLUSION

The proposed system has improved the movement in search of food of whale based on Levy distribution for classification of images of grape leaf diseases.

In this preprocessing, the proposed system utilizes convolution kernels to transform images into input data within the range (0,1). Subsequently, the levy distribution is incorporated into the Whale Optimization Algorithm model as an exploration search mechanism. A grape leaf dataset from the Plant Village project (www.plantvillage.org), comprising 4062 labeled images with dimensions of 256 by 256 pixels and divided into four classes, namely healthy, Black Rot, Black Measles, and Isariopsis leaf spot, is used to evaluate the performance of the proposed system. Experimental results demonstrate that the proposed system accurately recognize all five compared algorithms.

ACKNOWLEDGMENT

The authors would like to acknowledge School of Information and Communication Technology, University of Phayao, Thailand and Big Data Institute, Thailand for all resources and financial support.

REFERENCES

[1] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, and M. Woźniak, ‘‘A survey of deep convolutional neural networks applied for prediction of plant leaf diseases,’’ Sensors, vol. 21, no. 14, p. 4749, Jul. 2021.

[2] Ramanjot, U. Mittal, A. Wadhawan, J. Singla, N. Z. Jhanjhi, R. M. Ghoniem, S. K. Ray, and A. Abdelmaboud, ‘‘Plant disease detection and classification: A systematic literature review,’’ Sensors, vol. 23, no. 10, p. 4769, May 2023.

[3] G. A. Carlson, “A decision theoretic approach to crop disease prediction and control,” American Journal of Agricultural Economics, vol. 52, no. 2, pp. 216–223, 1970.

[4] N. Agrawal, J. Singhai, and D. K. Agarwal, ‘‘Grape leaf disease detection and classification using multi-class support vector machine,’’ in *Proc. Int. Conf. Recent Innov. Signal Process. Embedded Syst. (RISE)*, Oct. 2017, pp. 238–244, doi: 10.1109/RISE.2017.8378160.

[5] S. Nasiri and M. Z. Nejand, ‘‘A method based on image processing for the automatic diagnosis of grapevine leaf disease,’’ *Biosystem Eng. Iran*, vol. 53, no. 1, Apr. 2022, doi: 10.22059/ijbse.2022.327192.665432.

[6] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, ‘‘Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning,’’ *Smart Agricult. Technol.*, vol. 3, Feb. 2023, Art. no. 100081, doi: 10.1016/j.atech.2022.100081.

[7] K. Ashokkumar, S. Parthasarathy, S. Nandhini, and K. Ananthajothi, ‘‘Prediction of grape leaf through digital image using FRCNN,’’ *Meas., Sensors*, vol. 24, Dec. 2022, Art. no. 100447, doi: 10.1016/j.measen.2022.100447.

[8] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67.

[9] Chawla, M., & Duhan, M. (2018). Levy flights in metaheuristics optimization algorithms–a review. Applied Artificial Intelligence, 32(9-10), 802-821.

[10] Rayhan, Y., & Setyohadi, D. B. (2021, October). Classification of Grape Leaf Disease Using Convolutional Neural Network (CNN) with Pre-Trained Model VGG16. In 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) (pp. 1-5). IEEE.

[11] S. Nasiri and M. Z. Nejand, ‘‘A method based on image processing for the automatic diagnosis of grapevine leaf disease,’’ *Biosystem Eng. Iran*, vol. 53, no. 1, Apr. 2022, doi: 10.22059/ijbse.2022.327192.665432.

[12] A. Adeel, M. A. Khan, M. Sharif, F. Azam, J. H. Shah, T. Umer, and S.Wan, ‘‘Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion,’’ *Sustain. Comput., Informat. Syst.*, vol. 24, Dec. 2019, Art. no. 100349, doi: 10.1016/j.suscom.2019.08.002.

[13] M. A. Hasan, Y. Riyanto, and D. Riana, ‘‘Grape leaf image disease classification using CNN-VGG16 model,’’ *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 4, pp. 218–223, Oct. 2021, doi: 10.14710/jtsiskom.2021.14013.

turnitin Page 76 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Modification of Sand Cat Swarm Optimization for Classification Problems

Sakkayaphop Pravesjit, Krittika Kantawong, Sathien Hunta
School of Information and Communication Technology University of Phayao
Phayao, Thailand
sakkayaphop.pr@up.ac.th, krittika.ka@up.ac.th, sathien.hu@up.ac.th

Duangjai Jitkongchuen
Big Data Instutite
Bangkok, Thailand
duangjai.ji@bdi.or.th

Arit Thammano
Facluty of Information Technology,
King Mongkut's Institute of Technology Ladkrabang
Ladkrabang, Thailand
arit@it.kmitl.ac.th

Panchit Longpradit
Faculty of Social Sciences and Humanities
Mahidol University
Nakhon Pathom, Thailand
panchit.lon@mahidol.ac.th

Abstract— The proposed system represents an enhanced version in search of food of Sand Cat based on Levy distribution and Firework algorithm for the image classification of grape leaf diseases. In the preprocessing step, the proposed system utilizes convolution kernels to transform images into input data within the range of (0,1). Successively, the Levy distribution and Firework algorithm are incorporated into the SCSO model as an exploration search mechanism. The study employed a grape leaf dataset sourced from the Plant Village project (www.plantvillage.org), comprising 4062 labeled images measuring 256 by 256 pixels and categorized into 4 distinct classes: healthy, Black Rot, Black Measles, and Isariopsis leaf spot, which was utilized to evaluate the efficacy of the proposed system. The experimental findings demonstrate that the proposed system outperforms the analyses of VGG16, GLCM with SVM, Low contrast haze reduction-neighborhood component analysis with SVM, and SCSO.

Keywords—sand cat swarm optimization, levy distribution, convolution kernel, grape leaf disease, firework algorithm

I. INTRODUCTION

In the realm of image processing, digital image processing stands out as the most and widely popular technique. Within agricultural production field, the prevalence of plant diseases poses a significant challenge, adversely affecting both the quality and yield of crops [1]. This is particularly apparent in the case of grape plants, which are susceptible to diseases such as powdery mildew, brown blotch, and anthracnose, leading to notable reductions in grape yield and quality. Moreover, the diagnosis of plant diseases is frequently hindered by the limited expertise of farmers [2], who are often constrained by their professional knowledge and experience. For grapes, infected leaves typically exhibit visible spots, prompting farmers to either conduct visual inspections themselves or capture images of grape plant leaves to send out for experts in diagnosing the disease [3]. In consequence, expedient and precise identification and diagnosis of the disease pose a challenge.

Presently, artificial intelligence technology finds extensive application in agricultural plant disease classification, encompassing techniques such as computer vision, evolutionary algorithms, deep learning algorithms, and more, resulting in numerous studies within this promising field. That et al. presented the VGG16 based model for feature extraction for grape leaf disease classification model [4]. The structure of feature extraction employed deep network structure, comprising 16 convolution layers and 3 fully connected layers. Following that, SVM was then utilized as a model for classification of disease classes. Jain and Periyasamy adopted the VGG16 architecture to extract features from grape leaf images [5]. The Random Forest classifier can handle the multi-class classification task with high accuracy. Nasiri and Nejand [6] proposed a methodology for detecting grape leaf diseases, comprising segmentation using Gray-Level Co-occurrence Matrix (GLCM) and Gray Level Run Length Matrix (GLRLM), followed by classification utilizing Local Binary Pattern (LBP) based features. Another investigation conducted by Lauguico et al. introduced a transfer learning approach leveraging Alexnet based on Regions with Convolutional Neural Networks [7]. Ji et al. introduced a model that integrates GoogLeNet and ResNet architectures for the purpose of grape leaf disease detection [8]. This unified model was employed to classify healthy and diseased grape leaves, with a specific focus on identifying common diseases including Black Rot, esca, and Isariopsis leaf spot.

This paper presents an enhancement of Sand Cat Swarm Optimization Algorithm for classification of grape leaf disease. The performance of the proposed algorithm was evaluated and compared with VGG16, GLCM with SVM, GoogleNet &ResNet50, AlexNet, and SCSO.

Following this introduction, Section 2 provides an overview of the dataset. Sand Cat Swarm Optimization is delineated in Section 3, while Section 4 offers a description of the proposed algorithm. Section 5 presents and discusses upon the experimental results. Finally, Section 6 draws the conclusions derived from the studyDataset description

This study employed a publicly accessible dataset sourced from the Plant Village project (www.plantvillage.org). Specifically, the grape leaf dataset, consisting of 4062 labeled

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

images measuring 256 by 256 pixels, was selected. Subsequently, the dataset was partitioned into two subsets based on health status, distinguishing between healthy and disease specimens. Further categorization was performed to identify 3 distinct diseases: Black Rot, Black Measles, and Isariopsis leaf spot. The sample images for each class are presented in Table 1.

II. SAND CAT SWARM OPTIMIZATION ALGORITHM

The Sand Cat Swarm Optimization (SCSO) [9] algorithm, inspired by the natural behaviors of the sand cat, is characterized by two main actions: search for prey (exploration) and attacking the prey (exploitation). These processes are elaborated upon in detail as follows:

(i) search for prey (exploration): in cases where |R| >= 1, it explains the sand cat updates its position by considering the best-candidate position and its current position within its sensitivity range. Consequently, sand cats can explore potential alternative prey positions, as described in Eq. (1). This equation expands the search area between the current position and that of the prey, offering an additional opportunity for exploration.

$$sc_{t+1}^{c} = r * (sc_{t}^{bc} - rand(0,1) * x_{t}^{c}) \quad (1)$$

$$r = r_G * rand(0,1) \quad (2)$$

$$r_G = S_m - \left[\frac{S_m * iter_c}{iter_{Max}}\right] \quad (3)$$

$$R = 2 * r_G * rand(0,1) - r_G \quad (4)$$

where sc_t^c represents the current location of the sand cat and sc_{t+1}^c denotes the location of the sand cat's next moment. sc_t^{bc} signifies the location of the sand cat closest to the prey. r_G is a linear convergence factor and its value decreasing from 2 to 0 from the increasing number of iterations. $iter_c$ is the number of iterations in this present time, and $iter_{max}$ is the maximum of the iterations. $rand$ is a random value within the range of (0,1). The number is derived from the acoustic attributes of the sand cat, with an assumed value of 2, while the variable r represents sensitivity span exhibited by each cat.

TABLE I. SAMPLE IMAGES FOR EACH CLASS FROM THE PLANT VILLAGE DATASET.

Input class	Image samples
Healthy	
Black Rot	
Black Measles	
Isariopsis leaf spot	

turnitin Page 78 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

(ii) Attacking the prey (exploitation): Eq.(5) is used to determine the distance between the sand cat and its prey, simulating the sand cat's attack on the target. The circular sensitivity range is assumed for sand cats. Positions are randomly generated from the best and current positions. Subsequently, a random angle is selected using the roulette method, and the attack is executed using Formula (6). This approach effectively mitigates the risk of the algorithm succumbing to local optima by introducing randomness into the angle selection process.

$$sc_{rnd} = \left| rand(0,1) * sc_t^{bc} - sc_t^c \right| \quad (5)$$

$$sc_{t+1}^c = sc_t^{bc} - r * sc_{rnd} * \cos\theta \quad (6)$$

The Sand Cat Swarm Optimization Algorithm (SCSO) can efficiently solve the optimization problem, particularly for low-dimensional functions. However, it encounters limitations when applied to high-dimensional functions. The algorithm's performance notably declines and is susceptible to getting trapped in local optima due to its tendency to converge rapidly at the outset.

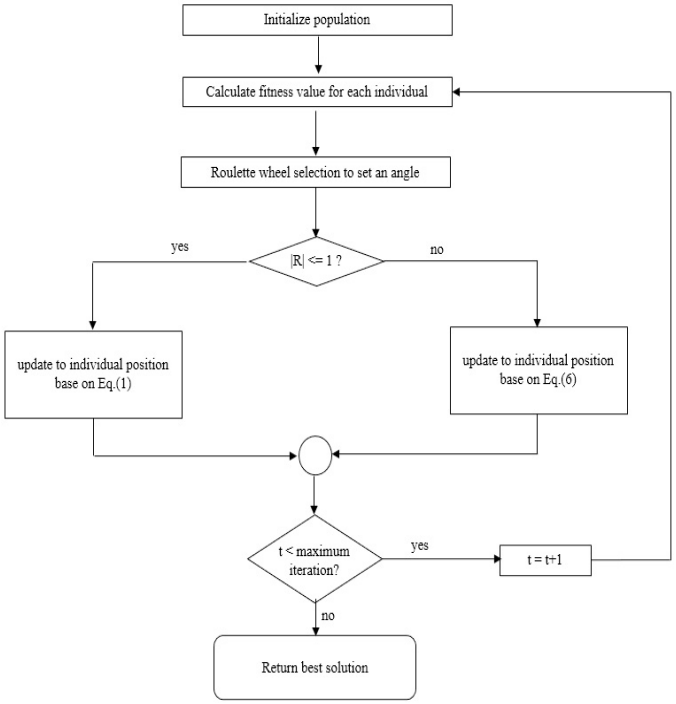


Fig. 1. The flowchart of sand cat swarm optimization algorithm.

III. THE PROPOSED ALGORITHM

This paper aims to enhance the exploration of position updating of Sand Cat Swarm Optimization algorithm for classification problems. The flowchart of the proposed algorithm is exhibited in Fig 2. After initializing populations and assessing their fitness values, the Levy distribution and Firework algorithm were applied to enhance the process of exploration of position updating. The processes are explained in detail as follows:

A. Levy distribution.

In this study, it employed random numbers through Levy Flight (LF), comprising 2 distinct phases. The initial phase entails the selection of a random direction, while the subsequent phase involves the generation of steps according to Levy Distribution (LD) [10]. Random direction selection is based on a uniform distribution. The steps are described as follows:

$$LD = \frac{u}{|v|^{1/\beta}} \quad (7)$$

where u and v are drawn from normal distributions. This implies that:

$$u = N(0, \delta_u^2, n, m) \quad (8)$$

$$\delta_u = \left\{ \frac{\tau(1+\beta)\sin(\pi\beta/2)}{\tau\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad (9)$$

$$\beta \text{ is constant value with } 0 < \beta < 2 \quad (10)$$

$$v = N(0,1,n,m) \quad (11)$$

B. Firework algorithm

The Fireworks Algorithm (FWA), inspired by the fireworks explosion simulation [11], [12], falls within the umbrella of the Swarm Intelligence algorithms category. The process of a firework explosion can be viewed as searching in the surrounding local area where the firework is set off through the sparks generated in the explosion. The steps are shown as follows:

$$F_{t+1}^i = F_t^i + U \quad (12)$$

$$A_i = \bar{A} * \left(\frac{f(F_t^i) - f(F_t^{best}) + \varepsilon}{\sum_{i=1}^n \left(f(F_t^i) - f(F_t^{best}) \right) + \varepsilon} \right) \quad (13)$$

$$S_i = Sm * \left(\frac{f(F_t^{worst}) - f(F_t^i) + \varepsilon}{\sum_{i=1}^n \left(f(F_t^{worst}) - f(F_t^i) \right) + \varepsilon} \right) \quad (14)$$

where Sm is the total number of sparks, $f(F_t^{worst})$ is the worst fitness value, , $f(F_t^{best})$ is the best fitness value, $\bar{A}$ is the desired sum of amplitudes, ε is a constant used to avoid zero-division operations, and U is adding a uniform random number in the interval $(-A_i, A_i)$.

Consequently, the populations of Levy distribution, the Firework and sand cat were selected from the Eq. (15).

$$sc_{t+1} = max_{fitness} \begin{cases} sc - r * sc_{rnd} * \cos\theta \\ LD - r * sc_{rnd} * \cos\theta \\ F - r * sc_{rnd} * \cos\theta \end{cases} \quad (15)$$

From Function (15), the process is to select the position of sand cat with the highest fitness value to use in this exploration step.

TABLE II. THE PARAMETERS USED IN THE EXPERIMENTS

Parameters	Values
Population size, NB	25
Dimension, (ND)	100
# of data class (n)	4
Max iteration	300
# of Data train	4062 – (# of Data test)
# of Data test	10% of each class data sets.

TABLE III. THE PARAMETERS USED IN THE LEVY DISTRIBUTION

Parameters	Values
Population size, NB	25
Dimension, (m)	100
Max iteration	300
# of data class (n)	4

TABLE IV. THE PARAMETERS USED IN THE FIREWORK ALGORITHM

Parameters	Values
Population size, NB	25
Dimension, (m)	100
Max iteration	300
# of data class (n)	4
Sm	5

IV. THE EXPERIMENTAL RESULTS

As can be seen in Figure 1, the grape leaf dataset comprised 4062 labeled images, each measuring 256 by 256 pixels. The grape leaf dataset was categorized into 4 classes, namely healthy, Black Rot, Black Measles, and Isariopsis leaf spot.

The computational results depicted in Table 5 provide a comprehensive comparison between the proposed algorithm and VGG16 from Yohan Rayhan et al. [13], GLCM with SVM from Sajjad Nasari et al. [14], Low contrast haze reduction-neighborhood component analysis with SVM by Alishba Adeel et al. [15], VGG16 by Arie Hasan et al. [16], and original SCSO from Seyyedabbasi and Kiani [9]. It suggests that the proposed algorithm has outperformed the others in comparison.

V. CONCLUSION

The proposed study represents an enhanced approach to food search of sand cats, utilising the Levy distribution and Firework algorithm for image classification of grape leaf diseases. In the preprocessing step, the proposed system uses convolution kernels to transform images into input data within the range of (0,1). Following this, the Levy distribution and Firework algorithm are integrated into the SCSO model to serve as an exploration search mechanism. The research employed a dataset of grape leaf images obtained from the Plant Village project (www.plantvillage.org). This dataset comprised 4062 labeled images measuring 256 by 256 pixels and was classified into 4 distinct categories: healthy, Black Rot, Black Measles, and Isariopsis leaf spot. These images were utilized to assess the effectiveness of the proposed algorithm. The experimental results has confirmed its superiority compared to analyses conducted using VGG16, GLCM with SVM, Low contrast haze reduction-neighborhood component analysis with SVM, and SCSO.

TABLE V. COMPARATIVE ANALYSIS OF THE PERFORMANCE OF THE PROPOSED WORK WITH THAT OF OTHER EXISTING WORKS.

Sl.no	Source	Methodology	Accuracy (in %)
1	Yohan Rayhan et al. [13]	VGG16	87.50
2	Sajjad Nasari et al. [14]	GLCM with SVM	89.93
3	Alishba Adeel et al. [15]	Low contrast haze reduction-neighborhood component analysis with SVM	91.34
4	Arie Hasan et al. [16]	VGG16	95.00
5	Seyyedabbasi and Kiani [9]	SCSO	86.76
6	Proposed algorithm	SCSO-LD-FWA	**99.27**

ACKNOWLEDGMENT

The authors would like to acknowledge School of Information and Communication Technology, University of Phayao, Thailand and Big Data Institute, Thailand for all resources and financial support.

REFERENCES

[1] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, and M. Woźniak, ‘‘A survey of deep convolutional neural networks applied for prediction of plant leaf diseases,’’ Sensors, vol. 21, no. 14, p. 4749, Jul. 2021.

[2] Ramanjot, U. Mittal, A. Wadhawan, J. Singla, N. Z. Jhanjhi, R. M. Ghoniem, S. K. Ray, and A. Abdelmaboud, ‘‘Plant disease detection and classification: A systematic literature review,’’ Sensors, vol. 23, no. 10, p. 4769, May 2023.

[3]] G. A. Carlson, “A decision theoretic approach to crop disease prediction and control,” American Journal of Agricultural Economics, vol. 52, no. 2, pp. 216–223, 1970.

[4] Thet, K. Z., Htwe, K. K., & Thein, M. M. (2020, November). Grape leaf diseases classification using convolutional neural network. In 2020 international conference on advanced information technologies (ICAIT) (pp. 147-152). IEEE.

[5] Jain, B., & Periyasamy, S. (2022). Grapes disease detection using transfer learning. arXiv preprint arXiv:2208.07647.

[6] S. Nasiri and M. Z. Nejand, ‘‘A method based on image processing for the automatic diagnosis of grapevine leaf disease,’’ *Biosystem Eng. Iran*, vol. 53, no. 1, Apr. 2022, doi: 10.22059/ijbse.2022.327192.665432.

[7] Lauguico, S., Concepcion, R., Tobias, R. R., Bandala, A., Vicerra, R. R., & Dadios, E. (2020, November). Grape leaf multi-disease detection with confidence value using transfer learning integrated to regions with convolutional neural networks. In 2020 IEEE region 10 conference (TENCON) (pp. 767-772). IEEE.

[8] Ji, M., Zhang, L., & Wu, Q. (2020). Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. Information Processing in Agriculture, 7(3), 418-426.

[9] Seyyedabbasi, A., & Kiani, F. (2023). Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. Engineering with Computers, 39(4), 2627-2651.

[10] Chawla, M., & Duhan, M. (2018). Levy flights in metaheuristics optimization algorithms–a review. Applied Artificial Intelligence, 32(9-10), 802-821.

[11] Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. In Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I 1 (pp. 355-364). Springer Berlin Heidelberg

[12] Bejinariu, S. I., Costin, H., Rotaru, F., Luca, R., Niţă, C. D., & Lazăr, C. (2018). Fireworks algorithm based image registration. In Soft Computing Applications: Proceedings of the 7th International Workshop Soft Computing Applications (SOFA 2016), Volume 1 7 (pp. 509-523). Springer International Publishing.

[13] Rayhan, Y., & Setyohadi, D. B. (2021, October). Classification of Grape Leaf Disease Using Convolutional Neural Network (CNN) with Pre-Trained Model VGG16. In 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) (pp. 1-5). IEEE.

[14] S. Nasiri and M. Z. Nejand, ‘‘A method based on image processingfor the automatic diagnosis of grapevine leaf disease,’’ Biosystem Eng. Iran, vol. 53, no. 1, Apr. 2022, doi: 10.22059/ijbse.2022.327192.665432.

[15] Adeel, A., Khan, M. A., Sharif, M., Azam, F., Shah, J. H., Umer, T., & Wan, S. (2019). Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion. Sustainable Computing: Informatics and Systems, 24, 100349.

[16] Hasan, M. A., Riyanto, Y., & Riana, D. (2021). Grape leaf image disease classification using CNN-VGG16 model. J. Teknol. dan Sist. Komput, 9(4), 218-223.

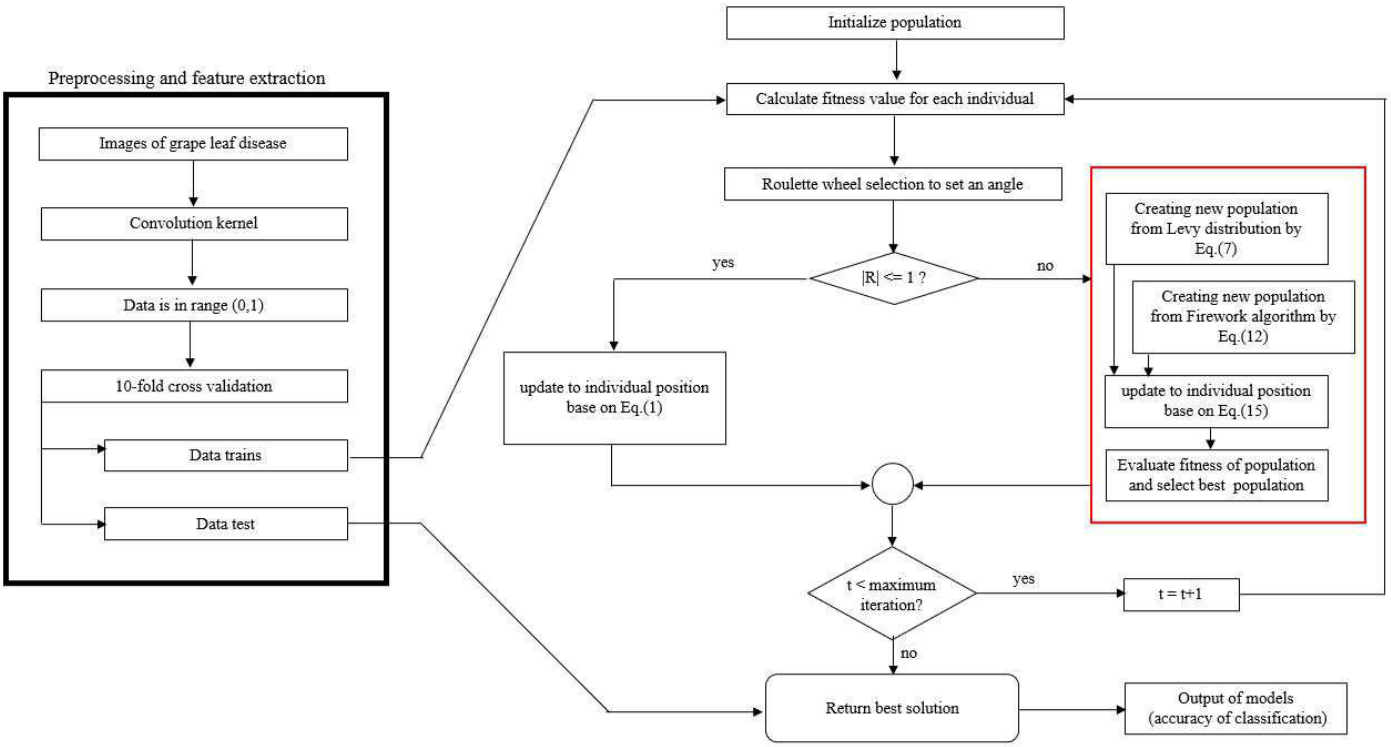


Fig. 2. The flowchart of the proposed algorithm.

turnitin Page 81 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Basking Behavior in Cold-blooded and Warm-blooded Reptiles: A Systematic Review of Interspecies Treatment

Chanatkit Harnnuengnit
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
chanatkit.har@dsil.ac.th

Sarochar Khambuo
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
sarochar.soo@dsil.ac.th

Puthyrom Tep
Darunsikkhalai School for Innovative Learning
King Mongkut's University of Technology Thonburi
Bangkok, Thailand
https://orcid.org/0000-0001-7647-1536

***Abstract*— This systematic review examines the possible use of basking behavior in both cold-blooded and warm-blooded species as a means of thermoregulation and its impact on interspecies care. Based on literature from 1960 to 2023, the review investigates the evolutionary link between cold-blooded and warm-blooded reptiles and their shared use of basking to regulate body temperature. The review followed a three-step guideline which involved planning, conducting, and reporting. After exclusions, a total of 14 articles were included for analysis. Results show that while warm-blooded reptiles are usually seen as fully endothermic, they also exhibit basking behavior. Likewise, cold-blooded animals depend on basking for thermoregulation, displaying adaptive behaviors to regulate their body temperature. The review highlights the importance of basking as a shared thermoregulatory method among cold-blooded and warm-blooded reptiles, indicating its potential for interspecies care.**

Keywords*— *Basking behavior, Warm-blooded, Cold-blooded, Thermoregulation, Systematic review, Interspecies treatment

I. Introduction

Animals have diverse ways of coping with illnesses, depending on their habitat, physiology, and genetics [1]. Animals in the same class can still react differently to diseases that others can get or avoid [2]. Some animals, such as macaws, use natural remedies like clay to neutralize toxins from their food [3]. Others, such as domesticated animals, rely on human intervention and veterinary care to provide them with suitable medication [4]. However, not all animals respond equally to the same treatment, even if they belong to the same class or share a common ancestry [5]. This poses a challenge for veterinarians who have to consider the specific needs and characteristics of each animal species [6].

The animal kingdom is composed of a diverse array of species that have evolved different strategies to cope with the thermal environment. One of the major distinctions among animals is whether they are cold-blooded or warm-blooded. Cold-blooded animals, also known as poikilotherms, are those that have a variable body temperature that depends on the external temperature [7]. Examples of cold-blooded animals include frogs, newts, lizards, and snakes.

These animals typically gain heat by basking in the sun or lose heat by immersing in water. In regions where the temperature changes seasonally, cold-blooded animals may seek shelter and enter a state of dormancy called brumation. Cold-blooded animals also tend to have lower metabolic rates and food requirements than warm-blooded animals [8].

Warm-blooded animals, also known as homoiotherms, are those that have a constant body temperature that is independent of the external temperature [9]. Examples of warm-blooded animals include mammals and birds. These animals can regulate their body temperature by adjusting their metabolic rate and heat production. They can also use behavioral and physiological mechanisms, such as shivering, sweating, and panting, to dissipate excess heat or conserve heat loss [10]. Warm-blooded animals can be easily distinguished from cold-blooded animals by their higher body temperature, which is usually around 37°C for mammals and 40°C for birds [9].

Cold-blooded and warm-blooded reptiles share a common evolutionary origin. They both descended from a group of reptiles called diapsids, which had two openings in the skull, about 350 million years ago [11]. Despite this long divergence, they still retain some physiological traits in common, such as having scales and laying eggs [12]. However, they also differ markedly in many aspects of their morphology and ecology, such as feathers, flight, and vocalization [13, 14]. These differences challenge the human cognitive tendency to categorize animals based on their superficial resemblance [15].

One of the most striking differences between cold-blooded and warm-blooded reptiles is their thermoregulatory strategy. Birds are often considered as warm-blooded or endothermic animals, meaning that they can maintain a constant body temperature regardless of the environmental conditions [16]. However, this is only partially true, as birds lack the brown adipose tissue that is responsible for heat production in most mammals [17]. Brown adipose tissue contains a high density of mitochondria, the cellular organelles that generate energy through oxidative metabolism [18]. In mammals, brown adipose tissue can produce up to 1200 watts of heat in a healthy human body [18].

Birds, on the other hand, do not have this tissue and rely on other mechanisms, such as shivering, to generate heat when needed [19]. Therefore, birds are not fully endothermic, but rather facultatively endothermic, meaning that they can switch between endothermy and ectothermy, or relying on external heat sources, depending on the situation [20]. This suggests that birds are not fully derived from their reptilian ancestors, but rather represent an intermediate stage of thermoregulatory evolution [21].

A fascinating question that emerges from this evolutionary relationship is whether cold-blooded and warm-blooded

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

reptiles (or avians and reptiles) can be treated with the same medication for certain diseases [22]. This could have practical implications for veterinary medicine, as it could provide an alternative option for treating animals when conventional methods are not available or effective [23].

One possible way to address this question is to investigate a common behavior that both cold-blooded and warm-blooded reptiles use for thermoregulation: basking. Basking is the process of exposing oneself to environmental heat sources, such as sunlight or heated rocks, to increase one's body temperature [24, 25]. Basking is known to be beneficial for cold-blooded animals, especially when they are sick or injured [26]. But can basking also help warm-blooded animals, especially birds, which are more closely related to reptiles than any other living group of animals?

This research aims to explore the feasibility of using basking as a treatment that is suitable for both cold-blooded and warm-blooded species. This research proposes that basking can reveal the phylogenetic and functional similarities and differences between these species and provide insights into their adaptation to different thermal environments.

It also can help both cold-blooded and warm-blooded species that have temperature-related illnesses, such as hypothermia and hyperthermia, by enhancing their immune system and metabolic rate. This study investigates the effects of basking on the health and well-being of both groups of animals, and the mechanisms and factors that enable or hinder interspecies treatment [27].

The objectives of this research are:

- To identify the physiological and behavioral responses of cold-blooded and warm-blooded species to basking in different thermal environments
- To identify the similarities between cold-blooded and warm-blooded species in terms of their adaptation to basking and their potential for interspecies treatment

This research has both academic and practical relevance. Academically, it will contribute to the understanding of the evolutionary and ecological connection between cold-blooded and warm-blooded reptiles. Practically, it will provide better care and treatment for these animals, especially for veterinarians and pet owners, by understanding the benefits and limitations of basking for cold-blooded and warm-blooded species. It will also educate the public about the importance of basking for animal welfare and conservation and encourage the cohabitation of different species that can benefit from each other's presence [28, 29]. Furthermore, it will enhance the knowledge and skills of veterinary students and other interested parties in dealing with various diseases and disorders that affect cold-blooded and warm-blooded reptiles. is [7].

II. METHODS

To conduct a reliable systematic review on this topic, the three-stage guideline provided by Tranfield, et al. [30] was followed. The three-stage guideline includes planning the review, conducting the review, and reporting and dissemination. During stage I, the need and goal of this review were identified based on the scoping study discussed in the previous section. During stage II, useful and high-quality papers were selected by performing a keyword search in various online bibliographic databases.

A. Scoping Review Procedure

This review focused on the literature published from 1960 to 2023. This is a relatively new and underexplored field, as most studies have assumed that birds are fully endothermic and do not need external heat sources to regulate their body temperature. Therefore, the number of articles that provided relevant and reliable information on this topic was limited.

B. Keyword Search

The following search terms were generated based on the definitions and synonyms of keywords (see Table 1). They were used to search for relevant literature on the topics of hypothermia, temperature regulation, basking, and interspecies therapy for birds and reptiles. When combining all keywords, the results received were not satisfactory, as the number of articles retrieved was low. Thus, two search terms were decided to use in searching using OR Boolean operator (see equation 1).

Hypothermia AND "Temperature regulation" AND Basking AND Bird = (A)

Hypothermia AND "Temperature regulation" AND Basking AND Reptile = (B) (1)

((A) OR (B))

TABLE I. DEFINITIONS AND SYNONYMS OF KEYWORDS

#	Concept	Definition	Synonym
1	Bird	Warm-blooded egg-laying animals called Aves possess wings, beaks, scaly legs, and lack teeth.	Avian, fowl, raptor, and fletchling
2	Reptile	Any animal belonging to the class Reptilia is an air-breathing vertebrate with internal fertilization, amniotic development, and skin scales covering some or all of its body.	Reptilian, Lizard, Reptilliform, and Saurian
3	Thermoregulation	Maintaining the core body temperature by balancing heat gain with heat loss.	Temperature Regulation, Heat Regulation, and Homeostasis
4	Basking	A common thermoregulatory behavior among many animals. It helps them enhance physiological performance. In adverse weather conditions, several terrestrial ectothermic animals regulate their body temperature by basking and seeking shelter in underground burrows.	Sunning and Sun-bathing
5	Hypothermia	It occurs when the body temperature drops below 35°C unintentionally. It can happen in moderate as well as extremely cold environments. Symptoms vary depending on the severity of hypothermia.	Cold exposure

The following databases were searched: Scopus, ScienceDirect, Sage, Wiley, Springer, and Taylor & Francis. Additionally, a separate Google Scholar advanced search was conducted. All duplicated articles found in Google Scholar were removed.

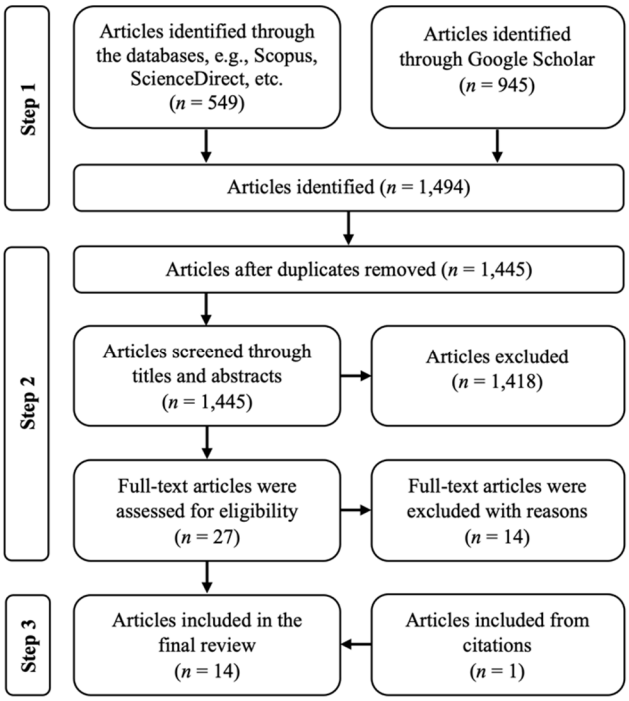


Fig. 1. Flowchart of the Screening Process

The search took place from August 24th to October 12th, 2023. The paper screening process was divided into three steps. The flowchart (see Figure 1) outlines the entire screening process.

C. Exclusion Criteria

The articles were excluded because they did not meet the following criteria: (1) the study subject was relevant to the theme or abstract of the article; (2) the article provided an explanation of the effects of basking on birds; (3) the article was not a duplicate or repetition of another article with the same author, title, and abstract; and (4) the article addressed how basking affects both reptiles and birds, not just thermoregulation in general.

III. Results

In step one, the final keywords were used to conduct searches across various databases. A total of 1,494 articles were found as shown in figure 2. In step two, the titles and abstracts of the articles from the first phase were screened. Certain criteria were applied to exclude the irrelevant ones. After the exclusion, a total of 27 articles remained as shown in figure 3.

In step three, the full-text articles were read to further eliminate the ones that were not related to the research question. There were 13 articles left after this step as shown in figure 4. This step includes:

- Synthesizing the Data: Combine the findings from the included studies to draw overall conclusions.
- Interpreting the Results: Discuss the implications of the findings, considering the quality and consistency of the evidence.

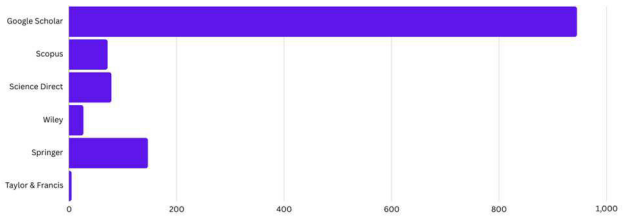


Fig. 2. Results of Step 1 - Final Keyword Search

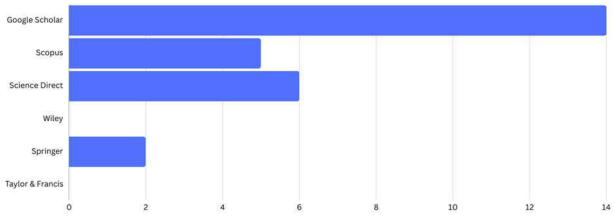


Fig. 3. Results of Step 2 - Title and Abstract Screening

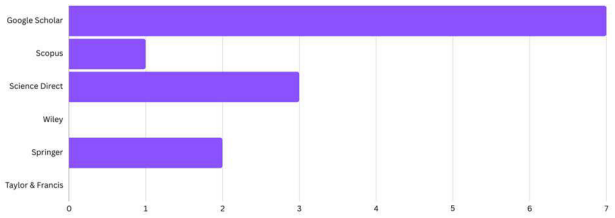


Fig. 4. Results of Step 3 - Full-Text Article Review

Subsequently, related articles were searched through citations within these 13 articles. Citation tracking is a recognized method in systematic reviews to identify additional relevant studies by following the references cited in the initial set of articles. One article was found and included in the final review. The remaining articles were condensed into a brief summary that highlighted the main findings and implications.

A total of 14 articles were acquired. The acquired articles are all journal articles, which mostly come from the USA, UK, and Australia. Table 2 presents the results of the article search and screening.

TABLE II. Results of Search and Screening

#	Database	Step 1	Step 2	Step 3
1	Google Scholar	945	14	7
2	Scopus	72	5	1
3	Science Direct	79	6	3
4	Wiley	27	0	0
5	Springer	`147	2	2
6	Taylor & Francis	5	0	0

In stage III, Table 3 provides detailed information about the author, the animal species studied, and a summary addressing the study gap mentioned earlier. This table also synthesizes findings to form comprehensive conclusions. Each summary was carefully analyzed to draw final conclusions, ensuring a thorough understanding of the findings, identifying key insights and implications.

TABLE III. DEMOGRAPHICS AND SUMMARY OF FINAL ARTICLES

#	Publication	Species	Summary
1	Kênia, et al. [31]	Avian	Birds and mammals often use sunning, basking, and social thermoregulation to control their body temperature. These behaviors can save energy by raising their body temperature by about 15–85%, depending on the animal species
2	Richards [32]	Reptile	The Galapagos marine iguana is a remarkable animal that can rapidly warm up in the sun after diving in the cold sea. This ability is crucial for maintaining its optimal body temperature and energy balance. The iguana's heart rate plays a key role in this process, as it increases during heating and decreases during cooling. This cardiovascular response is also observed in other reptiles that bask in the sun to regulate their temperature.
3	Jameson [33]	Avian	Basking is a behavior that some birds use to warm up in cold climates. It involves exposing the body to direct sunlight, which increases the temperature of both the feathers and the skin. Additionally, basking can reduce the metabolic rate and energy expenditure of birds, as they do not need to shiver or generate internal heat. Basking birds often seek shelter in tree hollows, where they can benefit from the heat radiation of the wood. The thermal properties of the wood depend on factors such as decay, water content, and structure. Basking, along with other adaptations such as fluffing up, huddling, caching, and tucking away, helps birds survive in harsh winter conditions.
4	Seebacher and Franklin [34]	NA	Basking is a common behavior in reptiles that involves sun exposure. Basking can have benefits beyond temperature regulation.
5	Glenn, et al. [35]	Reptile	Some studies suggest that reptiles and warm-blooded animals have similar brain cells that can detect temperature changes. For example, the horned lizard's head and body have a temperature gap of about 2-4 ◦C when it suns itself.
6	Regal [36]	NA	Reptiles like turtles, crocodiles, lizards, and snakes mostly get their body heat from external sources. Many of them bask in the sun, even if they live in areas where the rainforest covers the sky.
7	Brian G. Collins [37]	Avian	In the winter, the birds sun themselves in the sun when it is cold, usually in the beginning or end of the day. They do this to add to the heat they make inside their bodies.
8	Tan and Knight [38]	NA	Animals have various ways to control their body heat, such as huddling, extending, sunning, finding hot or cold places, making burrows or dens, and even spreading saliva. These ancient behaviors are seen in many kinds of animals, from fish to reptiles to birds and mammals that have warm blood.
9	Frederick [39]	NA	Reptiles like lizards mainly regulate their body heat by acting in certain ways. These actions, such as standing on two legs, burrowing in the sand, basking in the sun, and coming out partly in the morning, vary from small changes in position to bigger motions.
10	Aharon-Rotman, et al. [40]	Avian	Marsupial dunnarts need to use a heat lamp to warm up again, which shows how they control their body heat by using external sources, like the sun or a heat lamp.
11	Bereiter-Hahn, et al. [41]	Avian	Penguins show that birds can use sunshine to warm up instead of making heat inside their bodies. The color of their feathers, whether light or dark, can affect how much heat they get from the sun.
12	Davenport [42]	Reptile	Many animals that rely on external factors to control their body heat, such as sunbathing, can also adjust how fast they gain or lose heat. They do this by changing the blood circulation in their body.
13	Regal [25]	Reptile	An observed lizard had a body temperature of 310 degrees Celsius, while the air temperature was only 0 degrees Celsius. The lizard could control its body heat well in cold weather because of its basking activity.
14	Cowles and Bogert [43]	Reptile	Desert reptiles have a remarkable ability to control their body heat through behavior. They mainly achieve normal body temperatures by (1) choosing places in or on the ground, or rock, where they can get heat by direct contact, or (2) by sunning themselves, with all or some of their body exposed to the sun's heat. On the other hand, they avoid temperatures that are too high by (1) going to colder places, underground (by digging or using existing holes), in rock cracks, or under materials that keep heat out, (2) or by cooling down through breathing under very harsh conditions.

IV. DISCUSSION

This review aimed to determine whether basking, a common thermoregulation behavior, can be applied to both cold-blooded and warm-blooded species. The review focused on literature published from 1960 to 2023. Most studies have assumed that birds are fully endothermic and do not require external heat sources to regulate their body temperature; therefore, this field is relatively new and underexplored.

A. Physiological and Behavioral Responses to Basking

A recent article reviews literature from ancient to modern times, providing context for applying the findings from this review. The cross-over of articles suggesting that basking is suitable for cold-blooded and warm-blooded reptiles, along with the historical closeness in taxonomy and anatomy between the two groups, and current discussions on animal basking, support the idea that basking practices can be applied to them.

Some naturalists propose that birds have a direct evolutionary relationship with a group of animals with unique organs and structures, such as reptiles, based on their taxonomy and anatomy [44]. However, avians and mammals, which have different anatomies, both use basking to regulate

their body temperature. Basking is a widespread behavior that can save up to 85% of the energy needed to keep warm, depending on the species [31].

B. *Similarities between Cold-blooded and Warm-blooded Species*

Regarding warm-blooded animals, the efficiency of basking in avians is influenced by their feather color, structure, and function [41]. In winter, some avian, like the green heron, bask in the morning to increase their heat intake or recover their body temperature [37]. Others find shelter or rest in tree hollows [33], which have thermal advantages due to the bark, moisture, and wood decomposition [45].

On the other hand, cold-blooded animals control their body temperature through behavioral thermoregulation, such as basking in the sun or laying on a heated rock [39]. For example, the Galapagos marine iguana shows the significance of warming up using the basking process. This iguana utilizes its rapid-heating ability and solar irradiation to quickly reheat its body temperature to the preferred level of 37 degrees Celsius [32]. Ectotherms, like lizards and snakes, do not have full control over their body temperature, so they cannot maintain a stable temperature, but they still have control over their peripheral blood flow. Thus, they can manipulate the blood flow to exchange heat between their skin and surroundings [46], which means their body temperature mostly depends on their environment or changes with it.

These animal species also exhibit other secondary thermoregulation behaviors, such as curling up together, stretching out, bathing in the sun, searching for cold or warm locations, and digging tunnels or nests [38]. Some have more unique thermoregulation abilities. For instance, king quails and swallows tend to go into an inactive state and occasionally switch between ectothermic and endothermic states to conserve heat [35].

V. Conclusion

In conclusion, this review explored the possibility of applying basking, a common thermoregulation behavior, to both cold-blooded and warm-blooded reptiles. The review found that most studies have overlooked the role of external heat sources in regulating the body temperature of avian, which are assumed to be fully endothermic. However, the review also found evidence that cold-blooded and warm-blooded reptiles share a close evolutionary relationship, based on their taxonomy and anatomy, and that both groups use basking to conserve energy and maintain their preferred body temperature.

The review suggested that basking is a suitable and beneficial thermoregulation technique for both cold-blooded and warm-blooded species groups, and that more research is needed to understand the mechanisms and implications of this behavior.

VI. Implications & Limitations

This article faces several limitations due to the scarcity of recent research on the topic of thermoregulation in cold-blooded and warm-blooded species. Most of the existing studies that are relevant for this research provide overlapping or similar information, which creates data gaps that hinder the completion of the study. Another limitation is the lack of updated data in various databases; for instance, most of the data used in this analysis are from the 1970s and 1990s, with very few from the years 2010 to 2020. Therefore, some of the data may be outdated.

In terms of implications, this review article provides insights for veterinarians specializing in cold-blooded and warm-blooded reptiles, suggesting that basking can apply to both types. Basking can be utilized in treating temperature-related diseases such as hypothermia, hyperthermia, fever, or cold in both cold-blooded and warm-blooded reptiles. This is because basking under a heat source like the sun can help these animals regenerate their own heat while conserving energy in heat generation by around 80%.

Rather than expending all their energy to control their temperature to an ambient state while the disease is active and constantly fluctuating the body temperature of cold-blooded and warm-blooded reptiles, basking can be an effective natural method for maintaining optimal body temperature.

Acknowledgment

The first author would like to thank his parents, who supported him in writing this paper during the study and helped him in various ways, especially in researching and approving the article since the first few weeks of the study. The first author would also like to thank his advisors, who supported him in creating this article by approving the information acquired during the study, along with his parents. Finally, all authors would like to express their gratitude to Darunsikkhalai School for Innovative Learning for their support.

References

[1] T. Mirkena *et al.*, "Genetics of adaptation in domestic farm animals: A review," *Livestock Science,* vol. 132, no. 1, pp. 1-12, 2010, doi: 10.1016/j.livsci.2010.05.003.

[2] J. Haldane, "Disease and evolution," 1949.

[3] J. D. Gilardi, S. S. Duffey, C. A. Munn, and L. A. Tell, "Biochemical Functions of Geophagy in Parrots: Detoxification of Dietary Toxins and Cytoprotective Effects," *Journal of Chemical Ecology,* vol. 25, no. 4, pp. 897-922, 1999/04/01 1999, doi: 10.1023/A:1020857120217.

[4] B. Niemiec *et al.*, "World Small Animal Veterinary Association Global Dental Guidelines," *Journal of Small Animal Practice,* vol. 61, no. 7, pp. E36-E161, 2020, doi: 10.1111/jsap.13132.

[5] F. Bouchard and P. Huneman, From Groups to Individuals: Evolution and Emerging Individuality. MIT Press, 2013.

[6] D. G. Milchunas and W. K. Lauenroth, "Quantitative Effects of Grazing on Vegetation and Soils Over a Global Range of Environments," *Ecological Monographs,* vol. 63, no. 4, pp. 327-366, 1993, doi: 10.2307/2937150.

[7] R. W. Salt, "Cold and Cold-blooded Animals," (in eng), *Can J Comp Med Vet Sci,* vol. 13, no. 7, pp. 177-81, Jul 1949.

[8] C. M. Bogert, "How Reptiles Regulate Their Body Temperature," *Scientific American,* vol. 200, no. 4, pp. 105-120, 1959.

[9] A. Clarke and P. Rothery, "Scaling of body temperature in mammals and birds," *Functional Ecology,* vol. 22, no. 1, pp. 58-67, 2008/02/01 2008, doi: 10.1111/j.1365-2435.2007.01341.x.

[10] E. Mendelsohn, Heat and Life The Development of the Theory of Animal Heat. Harvard University Press, 1964.

[11] D. K. Griffin, D. M. Larkin, R. E. O'Connor, and M. N. Romanov, "Dinosaurs: Comparative Cytogenomics of Their Reptile Cousins and Avian Descendants," *Animals,* vol. 13, no. 1, p. 106, 2023.

[12] F. Seebacher, "A review of thermoregulation and physiological performance in reptiles: what is the role of phenotypic flexibility?," *Journal of Comparative Physiology B,* vol. 175, no. 7, pp. 453-461, 2005/10/01 2005, doi: 10.1007/s00360-005-0010-6.

[13] P. Brodkorb, "Origin and evolution of birds," *Avian biology,* vol. 1, pp. 19-55, 1971.

[14] J. H. Ostrom, "The Origin of Birds," *Annual Review of Earth and Planetary Sciences,* vol. 3, no. 1, pp. 55-77, 1975/05/01 1975, doi: 10.1146/annurev.ea.03.050175.000415.

[15] S. Atran, "Folk biology and the anthropology of science: Cognitive universals and cultural particulars," *Behavioral and brain sciences,* vol. 21, no. 4, pp. 547-569, 1998.

[16] H. G. Barbour, "THE HEAT-REGULATING MECHANISM OF THE BODY," *Physiological Reviews,* vol. 1, no. 2, pp. 295-326, 1921, doi: 10.1152/physrev.1921.1.2.295.

[17] F. J. García-García, A. Monistrol-Mula, F. Cardellach, and G. Garrabou, "Nutrition, Bioenergetics, and Metabolic Syndrome," *Nutrients,* vol. 12, no. 9, p. 2785, 2020.

[18] J. Pizzorno, "Mitochondria-Fundamental to Life and Health," (in eng), *Integr Med (Encinitas),* vol. 13, no. 2, pp. 8-15, Apr 2014.

[19] G. Grigg, J. Nowack, J. E. P. W. Bicudo, N. C. Bal, H. N. Woodward, and R. S. Seymour, "Whole-body endothermy: ancient, homologous and widespread among the ancestors of mammals, birds and crocodylians," *Biological Reviews,* vol. 97, no. 2, pp. 766-801, 2022, doi: 10.1111/brv.12822.

[20] A. C. Kirby, "A characterisation of the integumentary skeleton of lizards (Reptilia: Squamata)," UCL (University College London), 2020.

[21] J. L. Marx, "Warm-Blooded Dinosaurs: Evidence Pro and Con," *Science,* vol. 199, no. 4336, pp. 1424-1426, 1978, doi: doi:10.1126/science.199.4336.1424.

[22] J. E. Cooper and M. E. Cooper, *Introduction to veterinary and comparative forensic medicine*. John Wiley & Sons, 2008.

[23] W. V. Holt, "Mechanisms of sperm storage in the female reproductive tract: an interspecies comparison," *Reproduction in domestic animals,* vol. 46, pp. 68-74, 2011.

[24] B. H. Brattstrom, "Body temperatures of reptiles," *American Midland Naturalist,* pp. 376-422, 1965.

[25] P. J. Regal, "The Evolutionary Origin of Feathers," *The Quarterly Review of Biology,* vol. 50, no. 1, pp. 35-66, 1975, doi: 10.1086/408299.

[26] R. L. Ditmars, Reptiles of the world: tortoises and turtles, crocodilians, lizards and snakes of the Eastern and Western Hemispheres. Sturgis and Walton, 1910.

[27] M. V. A. Confessor, L. E. T. Mendonça, J. S. Mourão, and R. R. N. Alves, "Animals to heal animals: ethnoveterinary practices in semiarid region, Northeastern Brazil," *Journal of Ethnobiology and Ethnomedicine,* vol. 5, no. 1, p. 37, 2009/11/26 2009, doi: 10.1186/1746-4269-5-37.

[28] J. H.-D. Stephen, "Clinical Aspects of Reptile Behavior," *Veterinary Clinics of North America: Exotic Animal Practice,* vol. 4, no. 3, pp. 599-612, 2001, doi: 10.1016/S1094-9194(17)30025-7.

[29] S. P. Wensley, "Animal welfare and the human–animal bond: considerations for veterinary faculty, students, and practitioners," *Journal of Veterinary Medical Education,* vol. 35, no. 4, pp. 532-539, 2008.

[30] D. Tranfield, D. Denyer, and P. Smart, "Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review," *British Journal of Management,* vol. 14, no. 3, pp. 207-222, 2003/09/01 2003, doi: 10.1111/1467-8551.00375.

[31] C. B. Kênia, C. H. B. Renata, and G. S. B. Luiz, "Physiology of temperature regulation: Comparative aspects," *Comparative Biochemistry and Physiology Part A: Molecular & Integrative Physiology,* vol. 147, no. 3, pp. 616-639, 2007, doi: 10.1016/j.cbpa.2006.06.032.

[32] S. A. Richards, "Behavioural Thermoregulation," in *Temperature Regulation*, S. A. Richards Ed. Boston, MA: Springer US, 1973, pp. 110-142.

[33] E. W. Jameson, "Thermoregulation and Water Balance," in *Patterns of Vertebrate Biology*, E. W. Jameson Ed. New York, NY: Springer New York, 1981, pp. 146-186.

[34] F. Seebacher and C. E. Franklin, "Physiological mechanisms of thermoregulation in reptiles: a review," *Journal of Comparative Physiology B,* vol. 175, no. 8, pp. 533-541, 2005/11/01 2005, doi: 10.1007/s00360-005-0007-1.

[35] J. T. Glenn, C. Viviana, and C. S. Matthew, "Respiratory cooling and thermoregulatory coupling in reptiles," *Respiratory Physiology & Neurobiology,* vol. 154, no. 1, pp. 302-318, 2006, doi: 10.1016/j.resp.2006.02.011.

[36] P. J. Regal, *An analysis of heat-seeking in a lizard*. University of California, Los Angeles, 1968.

[37] G. C. a. S. P. Brian G. Collins, "Metabolism, thermoregulation and evaporative water loss in two species of Australian nectar-feeding birds (family Meliphagidae)," *Comparative Biochemistry and Physiology Part A: Physiology,* vol. 67, no. 4, pp. 629-635, 1980, doi: 10.1016/0300-9629(80)90252-2.

[38] C. L. Tan and Z. A. Knight, "Regulation of body temperature by the nervous system," *Neuron,* vol. 98, no. 1, pp. 31-48, 2018.

[39] D. K. Frederick, "Thermal reinforcement and thermoregulatory behaviour in the lizard Dipsosaurus dorsalis: An operant technique," *Animal Behaviour,* vol. 17, pp. 446-451, 1969, doi: 10.1016/0003-3472(69)90145-6.

[40] Y. Aharon-Rotman, G. Körtner, C. B. Wacker, and F. Geiser, "Do small precocial birds enter torpor to conserve energy during development?," *Journal of Experimental Biology,* vol. 223, no. 21, p. jeb231761, 2020.

[41] J. Bereiter-Hahn, A. G. Matoltsy, and K. S. Richards, *Biology of the Integument: invertebrates*. Springer Science & Business Media, 2012.

[42] J. Davenport, Environmental Stress and Behavioural Adaptation. Springer Dordrecht, 1985.

[43] R. B. Cowles and C. M. Bogert, "A Preliminary Study of the Thermal Requirements of Desert Reptiles. Bulletin of the American Museum of Natural History, Volume 83: Article 5. Raymond Bridgman Cowles , Charles Mitchill Bogert," *The Quarterly Review of Biology,* vol. 20, no. 2, pp. 170-170, 1945/06/01 1945, doi: 10.1086/394795.

[44] R. B. Cowles and C. M. Bogert, "A preliminary study of the thermal requirements of desert reptiles. Bulletin of the AMNH; v. 83, article 5," 1944.

[45] M. Paclík and K. Weidinger, "Microclimate of tree cavities during winter nights—implications for roost site selection in birds," *International Journal of Biometeorology,* vol. 51, no. 4, pp. 287-293, 2007/03/01 2007, doi: 10.1007/s00484-006-0067-2.

[46] J. Davenport, *Environmental Stress and Behavioural Adaptation*. Springer Netherlands, 2012.

turnitin Page 87 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Elevating Air Quality Forecasting: Integrating Hybrid Clustering Techniques with Long Short-Term Memory Networks

Irfan Fari Ramadhan
Department of Information Systems
Universitas Multimedia Nusantara
Tangerang, Indonesia
irfan.ramadhan@student.umn.ac.id

Samuel Ady Sanjaya
Department of Information Systems
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

***Abstract*— Air pollution is a pressing issue in many urban areas, including Jakarta, Indonesia, where it poses significant health and environmental challenges. This research addresses the air pollution problem in Jakarta by proposing a hybrid approach that combines K-Means and K-Medoids clustering algorithms with a Long Short-Term Memory (LSTM) predictive model to forecast Air Pollution Standard Index (ISPU) datasets. Developed using data mining methodologies and the CRISP-DM framework, the hybrid model is implemented in stages, with the hybrid clustering method effectively reducing the root mean square error (RMSE) score, thereby improving model accuracy. To determine optimal clusters, the study conducted several iterations of clustering, measuring cluster distance using both Euclidean distance and Dynamic Time Warping (DTW). Systematic RMSE-based comparisons of predictive results were performed to identify the most accurate models, highlighting the significant influence of the optimizer choice on performance. Notably, the LSTM univariate results for the cluster 2 ISPU dataset from 2021 to 2023 consistently demonstrated the lowest RMSE scores, ranging from 2.91984 to 9.53943, indicating its effectiveness as the preferred model configuration. This hybrid approach presents a robust solution for improving air quality forecasting accuracy, offering valuable insights for mitigating air pollution in urban environments like Jakarta.**

Keywords—Air Quality, Dynamic Time Warping (DTW), K-Means, K-Medoids, LSTM

I. INTRODUCTION

The issue of air pollution and air quality has become a hot topic nearing the end of 2023. During the period from July to August 2023, more than 34 thousand social media netizens on Twitter expressed complaints regarding the condition of air pollution, with 85% of the discourse originating from Java Island [1]. Air pollution causing worsening air quality in DKI Jakarta isn't a new problem. In fact, from 2018 to 2022, the average concentration of PM 2.5 particulates in the province was 7 to 10 times higher than what the World Health Organization (WHO) recommends [2]. The air quality index is obtained through the measurement of air pollutant concentrations such as carbon monoxide and others. Higher concentrations indicate poorer air quality. In addition to the air quality index measurements, there is also the Air Pollution Standard Index (ISPU) established by the Ministry of Environment and Forestry (KLHK) of Indonesia. The measurements conducted by the Jakarta Provincial Government itself adhere to the provisions of Minister of State for the Environment Decision No. 45 of 1997, as outlined in Article 3 regarding the use of the Air Pollution Standard Index (ISPU) as informational material for the public regarding ambient air quality at specific locations and times, and in Article 4 regarding the list of parameters for the ISPU, consisting of particulates (PM10), carbon monoxide (CO), sulfur dioxide (SO2), nitrogen dioxide (NO2), and ozone (O3) [3].

Then, the government issued Minister of Environment and Forestry Regulation No. 14 of 2020, adding particulate matter (PM2.5) and hydrocarbons (HC) as parameters monitored with the help of Ambient Air Quality Monitoring Stations (SPKUA) [4]. Hence, it's necessary to apply data mining with clustering methods to find patterns in air pollution in DKI Jakarta using Air Pollution Standard Index (ISPU) data. To uncover hidden patterns in DKI Jakarta ISPU data and categorize data based on Ambient Air Quality Monitoring Stations (SPKU) placed in different areas, clustering methods are needed as they can identify temporal patterns in air pollutants [5]. Additionally, clustering is performed due to the large amount of data, which is not suitable for traditional analysis, and the obtained information will be more meaningful [6]. Then, the clustering method is combined using a hybrid approach with the Long Short-Term Memory (LSTM) neural network model as a prediction model for air pollutant concentration from ISPU data.

Clustering method itself is a technique to process data and divide it into groups or clusters based on specific patterns [7]. Furthermore, the implementation of LSTM itself is a part of neural networks that can process more complex and interconnected data compared to machine learning models like clustering because it utilizes the concept of the human brain in information processing [8]. In this research, a hybrid combined model of K-Means and K-Medoids will be utilized, resulting in final clusters, which will then be predicted using Long Short-Term Memory (LSTM). The purpose of this research extends beyond merely predicting pollutant concentrations from DKI Jakarta's ISPU data; it also aims to identify the optimal predictive models based on specific evaluation criteria for each optimizer used. The primary dataset consists of the Air Pollution Standard Index (ISPU) data, sourced from the Jakarta Environmental Agency, covering a comprehensive period from 2010 to 2023. This extensive temporal range allows for robust analysis and model training. In addition to the ISPU data, the research incorporates supporting weather data, which includes crucial variables such as temperature, air humidity, and wind speed. These weather variables influence air quality and, therefore, enhance the predictive capabilities of the models.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

II. Related Work and Literature Review

A. Related Work

As this study employs Hybrid Clustering with LSTM to predict the Air Pollution Standard Index (ISPU) variable in DKI Jakarta, previous studies serve as references regarding similar topics. In [9], the authors conducted clustering for air pollution in DKI Jakarta with an accuracy rate of the K-Means algorithm based on Logistic Regression testing on clusters of moderate air quality and unhealthy air quality, achieving 0.9622 or 96.22%. In [10] a hybrid model of K-Means with LSTM, out of 60 Ambient Air Quality Monitoring Stations, 2 clusters were formed, with the first cluster in developing areas and the second cluster in suburban areas. Despite providing poorer results than the univariate LSTM model due to various factors such as different meteorological conditions at each monitoring station location, the hybrid model's prediction results can reveal daily average PM10 concentration patterns. However, the hybrid model has a shorter training time. Therefore, the hybrid model can be more competitive and suitable for direct application in air quality forecasting. In [11], the LSTM model showed strong performance in predicting three parameters: temperature, humidity, and ISPU. The RMSE values of the predictions were smaller than the standard deviation of the test dataset. When predicting with four parameters, the best-performing ones were humidity, temperature, ISPU, and PM10.

Next, in [12], the forecasts made one hour ahead using BPNN, CNN, LSTM, and the hybrid CNN-LSTM model showed that LSTM's utilization as a model is considered optimal for forecasting several hours ahead. Generally, LSTM and the hybrid CNN-LSTM model outperformed CNN or BPNN. Furthermore, for the reference in forming the hybrid clustering model, a study [13] was utilized where the hybrid model of K-Means and PAM, also known as K-Medoids, was employed to predict the COVID-19 status based on data from 400 clinic patients in Iraq using a questionnaire. The final outcome indicated that the hybrid K-Means and PAM, or K-MP, model was more efficient and effective in identifying patient status compared to using either K-Means or PAM alone. In the study [14], which utilized the K-Medoids algorithm to implement time series clustering on cooking oil prices data across 34 provinces in Indonesia from October 2017 to October 2022, it was found that the optimal K value was 2 clusters based on a silhouette coefficient of 0.19. With Dynamic Time Warping (DTW) distance, there were 19 provinces in cluster 1, indicating cooking oil prices below cluster 2, while 15 provinces were in cluster 2, representing the highest cooking oil prices. Furthermore, in [15], the Long Short-Term Memory algorithm was employed to predict air temperature in Indonesia. Two optimizers were utilized: Adam and SGD. The evaluation results using the Adam optimizer showed an R2 accuracy of 32%, an MAE of 0.0068, and an RMSE of 0.99. The conclusion drawn was that the Adam optimizer outperformed SGD in predicting air temperature.

In this study, the configuration from [10] will be used for LSTM, while the study [13] will serve as a reference for creating the hybrid k-means with k-medoids model. Additionally, the study [15] will be consulted for the implementation of two different optimizers on LSTM, namely Adam and SGD. Other studies such as [9], [11], [12], [14], [16], [17], and [18] will be utilized as supplementary information in this research.

III. Methodology

This research will focus on examining the Air Pollution Standard Index (ISPU) data of DKI Jakarta from the period of 2010 to 2023. The precise timeframe in the dataset starts from January 1, 2010, to November 30, 2023.

A. CRISP-DM Workflow

In this study, one of the data mining methods, namely CRISP-DM, will be utilized, as depicted in Fig. 2. The following are the steps involved in this process:

1) Business Understanding

Through this research, the goal is to predict pollutant concentrations using Air Pollution Standard Index (ISPU) data from DKI Jakarta from 2010 to 2023. Additionally, there are variations in the dataset, such as dividing it into the periods 2010 – 2023 and 2021 – 2023. Another objective is to visualize the best model based on the evaluation metrics between two different optimizers. Furthermore, this study can also provide insights into the implementation of the hybrid model of K-Means and K-Medoids with LSTM to predict ISPU data from DKI Jakarta spanning from 2010 to 2023, supplemented by weather data such as temperature, air humidity, and wind speed.

2) Data Understanding

The next step is to conduct data understanding to comprehend the data so that the research objectives can be further processed and achieved. In this study, the data to be understood includes the attributes present in the Air Pollution Standard Index (ISPU) dataset of DKI Jakarta from 2010 to 2023. Additionally, there are several supporting weather-related attributes, namely temperature, humidity, and wind speed. The data is retrieved from the website https://satudata.jakarta.go.id/, while the supporting data is obtained from https://www.visualcrossing.com/weather-data.

3) Data Preparation

During the data preparation stage, the data to be used undergoes checking and inspection. The raw dataset retrieved is cleaned first from inappropriate values. Then, imputation or filling of null values is performed using specific methods such as moving average. In one study, the moving average and interpolation imputation methods demonstrated strong performance in imputing similar data, specifically for data containing ISPU variables [31]. After that, several columns are selected, and the dataset is merged as needed for the research, primarily as input for the modeling stage. The final step involves conducting exploratory data analysis (EDA) to further examine the characteristics of variables, such as checking data stationarity and selecting variables with the highest correlation to create a multivariate model using LSTM.

4) Modeling

In the modeling stage, a model will be created based on the selected algorithms. This research will utilize a hybrid model combining K-Means and K-Medoids, followed by prediction using Long Short-Term Memory (LSTM), as per previous research references.

The clustering process begins with K-Means, where the optimal number of clusters is determined using the Davies Bouldin Index. Following this, the algorithm identifies the smallest centroid distance from the K-Means output and

proceeds to apply K-Medoids based on this minimal distance. The resulting clusters serve as the foundation for the predictive model. Additionally, the LSTM prediction model employs various dataset configurations, including univariate and multivariate setups. Univariate modeling utilizes a single input variable, whereas multivariate modeling incorporates two variables with significant correlation, enhancing predictive accuracy. Furthermore, for the LSTM prediction model, optimization is conducted using both Adam and SGD algorithms. Additionally, the optimizers to be used in the LSTM prediction model are Adam and SGD.

Long Short-Term Memory, abbreviated as LSTM, is a type of Recurrent Neural Network (RNN) developed to overcome the limitations of RNNs, such as the vanishing and exploding gradient problem [26].

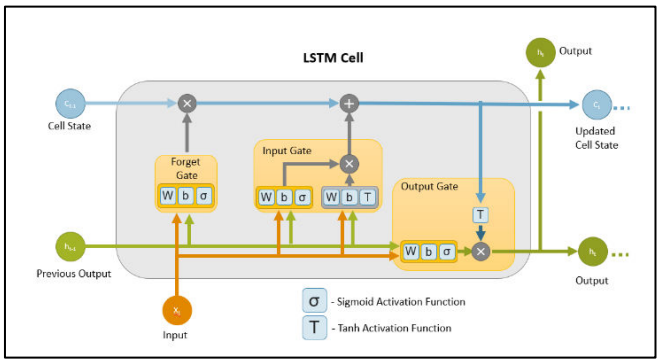


Fig. 1. LSTM Architecture [27]

In Fig. 1, it illustrates the architecture of Long Short-Term Memory (LSTM), which comprises three layers: input, hidden, and output. In the hidden layer, there are three gates: the Forget Gate, Input Gate, and Output Gate [28]. Each gate has its own specific task; for instance, the Forget Gate is responsible for determining which information to discard and how much information to discard from the cell state. The formula for the Forget Gate is as follows in (Equation 1 and Equation 2) [27]:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \quad (1)$$

$$c_f = c_{t-1} * f_t \quad (2)$$

Next, the Input Gate is useful for amplifying the results between two functional units and adding these results to the cell state. Below are (Equation 3, 4 and 5), which represent the formula for the Input Gate:

$$\tilde{c}_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (3)$$

$$m_t = \sigma(W_m.[h_{t-i}, x_t] + b_m) \quad (4)$$

$$c_t = c_f + \tilde{c}_t * m_t \quad (5)$$

Here are (Equation 6 and 7), representing the formula for the Output Gate:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(c_t) \quad (7)$$

5) Evaluation

The next step involves evaluating the results produced by the created models. In this study, the outcomes of the hybrid model will be assessed based on the final model, which is the prediction using LSTM. The evaluation metric used is the root mean square error (RMSE). Then, for each dataset and optimizer, comparisons will be made to identify which one yields the smallest RMSE value, serving as the criterion for determining the best model outcome. Root Mean Square Error, or RMSE, is a form of evaluation in a prediction model by summing the squared errors or differences between actual values and predicted values, dividing by the number of forecasted data points, and then taking the square root [29]. The following (Equation 8) is the formula for RMSE:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{n}} \quad (8)$$

The accuracy of the RMSE value is considered high if the resulting value is low; conversely, if the resulting value is high, the accuracy level decreases [30].

IV. CRISP-DM Process

As previously noted, this study will employ the CRISP-DM data mining framework to achieve its ultimate research objectives, e.g., predicting pollutant concentrations using DKI Jakarta's Air Pollution Standard Index (ISPU) data.

A. Business Understanding

In recent months, spanning from July to August 2023, social media users discussed the issue of air pollution and poor air quality in DKI Jakarta. This issue has become an annual concern due to worsening air pollution and quality in the region. The deteriorating air quality has even placed DKI Jakarta at the top of the list of cities with the worst air quality index. Another concern is the rise in cases of Acute Respiratory Infection (ARI) associated with air pollution and air quality in DKI Jakarta. Hence, there is a need for research to elucidate the patterns of poor air quality in DKI Jakarta. This study employs data mining methods and predictive modeling related to the Air Pollution Standard Index (ISPU) dataset. The ISPU data contains measurements of air quality applicable to Indonesia as per government regulations. By utilizing this data, a data mining model will be developed using hybrid clustering with the K-Means and K-Medoids algorithms to identify patterns emerging per Air Quality Monitoring Station (AQMS). Subsequently, the clustering results will be predicted using the long-short term memory (LSTM) neural network algorithm. The model's outcomes will be selected based on the smallest root mean square error (RMSE) value among the two optimizers used.

B. Data Understanding

In this study, the data utilized comprises the Air Pollution Standard Index (ISPU) data of DKI Jakarta spanning from 2010 to 2023. This dataset is daily and encompasses various air pollutants converted into ISPU values at each Air Quality Monitoring Station (AQMS), ranging from DKI 1 to DKI 5. The dataset includes variables named periode_data, tanggal, stasiun, pm10, pm25, so2, co, o3, no2, max, critical, and category. However, for data from 2010 to 2020, there is no pm25 variable or PM 2.5 pollutant due to the fact that it was only added as an ISPU parameter in the Regulation of the Minister of Environment and Forestry Number 14 of 2020. The following is Table 1 which describes the ISPU variables used in this study.

TABLE 1. ISPU VARIABLE EXPLANATION

Variable	Definition
Periode_data	Year and month period in ISPU data
tanggal	Observation date of ISPU data
stasiun	Name of air quality monitoring station (SPKU) measuring ISPU data observations
pm10	ISPU parameter in the form of particulate matter less than 10 micrometers
pm25	ISPU parameter in the form of particulate matter less than or equal to 2.5 micrometers
so2	ISPU parameter in the form of SO2 (sulfur dioxide)
co	ISPU parameter in the form of CO (carbon monoxide)
o3	ISPU parameter in the form of O3 (ozone)
no2	ISPU parameter in the form of NO2 (nitrogen dioxide)
max	Value or number of the highest ISPU parameter per data row
critical	Name of ISPU parameter with the highest value per data row
category	Category scale based on the highest ISPU value per data row

The variable that will not be used in the ISPU dataset for this research is "periode_data" because it can be replaced by the "tanggal" variable. Additionally, the locations of the air quality monitoring stations are: Central Jakarta, North Jakarta, South Jakarta, East Jakarta, West Jakarta. This study will divide the data range for clustering using data from 2010 to 2020. The ISPU data from 2010 to 2020 can also be used as training data, while for the years 2021 to 2023 or the remaining data, they will be used for prediction using long-short term memory (LSTM). Additionally, there is supporting weather data used in this research.

For the weather dataset, the configuration for the location in Central Jakarta is at latitude -6.20008 and longitude 106.833, with a maximum distance of 20 km from the air weather station. This weather dataset also includes the datetime variable, allowing it to be merged with the ISPU dataset. The selected variables in the weather data are temperature, humidity, and windspeed. These variables are chosen because other variables are considered less influential on the ISPU data, and to keep the number of variables used in modeling manageable.

C. *Data Preparation*

The data preparation stage involves processing the data or dataset to align it with the research objectives, especially in the modeling stage. The ISPU and weather datasets that have been retrieved need further processing, such as renaming columns, selecting columns to be used, imputing null values, and merging datasets from 2010 to 2023 and 2021 to 2023. This processing is necessary because the retrieved datasets are still raw and need to be cleaned to prevent errors during modeling. The data preparation process in more detail involves adjusting data types and renaming columns to facilitate the data merging process. Subsequently, null values are handled using methods such as moving average, interpolation, and backward fill. The final step of data preparation is to merge datasets for the years 2010 to 2023 and datasets with the PM2.5 variable for the years 2021 to 2023. This step is done to prepare for the modeling stage in the next phase.

Before proceeding to the modeling stage, exploratory data analysis (EDA) is conducted to examine the characteristics of the data or variables to be used as input for the model. In this EDA, correlations between variables are examined. Strong correlations are observed in the dataset for the years 2021 - 2023 between the PM10 and PM2.5 variables, as well as between temperature and humidity as weather variables. Furthermore, for the years 2010 - 2023, the ISPU parameters contributing as critical values are O3 and PM10, while for the years 2021 - 2023, they are PM2.5 and PM10. Additionally, the data's stationarity is checked, with the conclusion that the overall ISPU parameter data is stationary.

D. *Modeling*

1) Clustering Model and Evaluation

The first step in model creation involves using clustering algorithms. The algorithms selected are K-Means and K-Medoids, utilizing a hybrid model approach where a 2-step clustering process is employed with both algorithms. The initial clustering is performed using the K-Means algorithm to determine the optimal number of clusters based on evaluation using the Davies-Bouldin Index [32]. Subsequently, processing is conducted using K-Means based on the optimal clusters, and then the minimum distance is determined. Once the minimum distance is identified, K-Medoids is applied as the final clustering outcome.

The selected variables for the clustering process are ISPU parameters such as pm10, so2, co, and others. Variables from the weather dataset will not be utilized in the clustering modeling as they only serve as supportive data for future predictions. Additionally, data from the weather dataset consistently holds the same values for each day and does not reference the station columns present in the ISPU dataset.

cluster	2	3	4	5	6	7	8	9	10
dbi_scores_2_euclidean	0.847145	1.185644	1.429036	1.367789	1.368278	1.428539	1.379736	1.314876	1.341861
dbi_scores_3_dtw	0.827574	1.132869	1.374303	1.449223	1.815202	1.824176	1.895188	1.811876	2.099240
dbi_scores_4_euclidean	0.827456	1.066642	1.343638	1.393902	1.436134	1.504015	1.459818	1.468479	1.478901
dbi_scores_5_euclidean	0.971734	1.267244	1.399021	1.486853	1.269433	1.356770	1.412966	1.414717	1.434132
dbi_scores_6_dtw	0.972242	1.243408	1.391187	1.543341	1.524687	1.642197	1.343006	1.387928	1.496076

Fig. 2. Davies Bouldin Index Score

In Fig. 2, an optimal cluster check was conducted with 10 iterations, and cluster distance was measured using both Euclidean distance and Dynamic Time Warping (DTW). However, for the purpose of this study, the prediction model input will utilize DTW distance measurement because the cluster labels provided showed no difference from those obtained using Euclidean distance. Furthermore, as depicted in Fig. 3, the optimal cluster result consists of 2 clusters due to having the smallest DBI score. Subsequently, these clusters group the air quality monitoring stations (SPKU), both for the periods 2010 – 2023 and 2021 – 2023.

Fig. 3 depicts the final visualization of the clustering results for the ISPU dataset from 2021 to 2023. There are 2 clusters, with cluster 1 containing 3 stations: DKI3, DKI4, and DKI5. Cluster 2 groups stations DKI1 and DKI2. There is a difference in the number of stations, as seen in station DKI4 in cluster 1, which is due to the smaller amount of data available in the ISPU dataset for 2022. Additionally, when compared to the dataset from 2010 to 2023, which was split into training and test sets earlier, station DKI3 is now in cluster 1 for the 2021 to 2023 ISPU dataset, whereas previously it was in cluster 2.

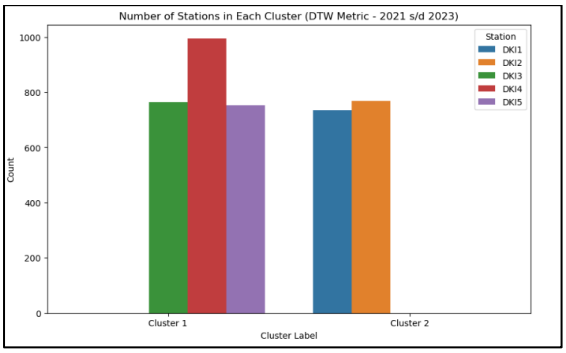


Fig. 3. Cluster Result ISPU 2021 – 2023

Fig. 4 is a visualization displaying the number of categories for the cluster results in the ISPU 2021 – 2023 dataset. The number of MODERATE and UNHEALTHY categories is higher in cluster 1, while cluster 2 tends only towards MODERATE. This indicates that in cluster 1, there are stations with the highest and most numerous ISPU values, resulting in unhealthy air quality around those station areas.

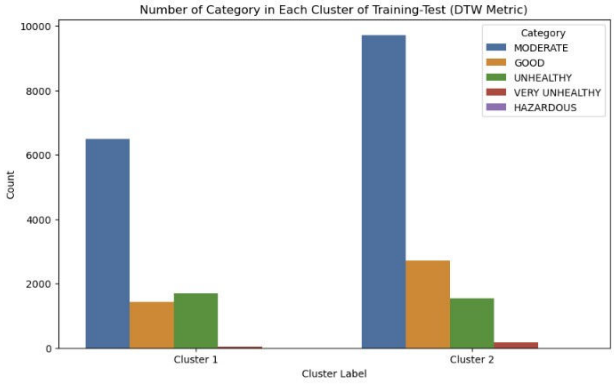


Fig. 4. Number of Category in ISPU 2021 – 2023 Clusters

2) Prediction Model

The next step in model development involves using LSTM to predict several columns or variables from the dataset. There are variations in the dataset, namely using ISPU 2010 – 2023 and ISPU 2021 – 2023, each divided into training and testing sets. Then, the clustering results for each dataset, ISPU 2010 – 2023 and ISPU 2021 – 2023, are used. Next, LSTM models are created in both univariate and multivariate forms. The Univariate model utilizes all ISPU parameters, while the multivariate model focuses on PM10 and PM2.5 from 2021 – 2023 due to their high correlation. The weather dataset is utilized in its entirety, without segmentation based on clustering results. Additionally, the multivariate model only employs data from 2021 – 2023, with temperature and air humidity variables. The following Table 2 is the LSTM configuration for this study.

TABLE 2. LSTM CONFIGURATION

Hyperparameter	Configuration
Number of layers	5
Dropout rate	0.1
Activation function	Tanh
Recurrent activation	Sigmoid
Loss function	MSE (Mean square error)
Optimizer	Adam & SGD

Based on Table 2, There is a distinction in this study, as it will utilize two optimizers, namely Adam and SGD, to determine the optimizer that yields the best RMSE value. Then, the training and testing are divided with a ratio of 80:20 (80% training, 20% testing), and for the timestep as input to the LSTM model, it consists of 7 observations back. The configurations made for the univariate and multivariate models are the same; the difference lies only in the number of inputs, where in the univariate model, there is 1 input for prediction resulting in 1 output, while in the multivariate model, 2 inputs are used for prediction resulting in 1 output.

E. Evaluation

The evaluation stage is the next step after the modeling process is completed. This stage involves evaluating the predictions made by the LSTM model because the clustering model has already been developed and explained earlier. In the evaluation phase, the focus lies on a metric called root mean square error (RMSE). The comparison of RMSE values across different datasets aims to identify the best parameters and model, seeking the smallest RMSE, which indicates superior performance. Additionally, comparisons are made between RMSE values obtained using different optimizers, such as Adam and SGD. Ultimately, the selection of ISPU parameters and optimizer relies on identifying the one with the smallest RMSE value.

TABLE 3. RMSE SCORE FOR UNIVARIATE ISPU 2021 – 2023

Variable	Optimizer	
	Adam	SGD
PM10	12.10330	**11.96749**
PM2.5	20.84967	**20.54647**
SO2	**12.54300**	12.99665
CO	7.18525	**6.80942**
O3	**12.65518**	12.93780
NO2	9.78611	**8.40968**
Temperature	**0.29568**	0.37026
Humidity	**1.73948**	2.11820
Wind Speed	**2.77726**	3.70842

Table 3 presents the comparison of RMSE values between optimizers using the ISPU 2021 – 2023 dataset variations. The LSTM model is also conducted in a univariate manner. The green-highlighted columns in Table 4 indicate the smallest RMSE values among the optimizer comparisons. Variables with the smallest RMSE values obtained from the Adam optimizer include SO2, O3, temperature, humidity, and wind speed. Conversely, variables PM10, PM2.5, CO, and NO2 achieve the smallest RMSE values when using the SGD optimizer.

TABLE 4. RMSE SCORE FOR UNIVARIATE ISPU 2021 – 2023 CLUSTER

Variable	Optimizer			
	Cluster 1		Cluster 2	
	Adam	SGD	Adam	SGD
PM10	14.55295	14.02638	**7.51523**	7.66060
PM2.5	26.07317	25.15387	**9.53943**	10.32629
SO2	12.59491	13.43460	**9.39729**	10.90920
CO	7.51756	7.27617	3.05594	**2.91984**
O3	16.76297	16.19257	7.89121	**6.98021**
NO2	6.37044	6.34439	**6.34293**	6.51062

Table 4 presents the comparison of RMSE values for univariate LSTM models on ISPU 2021 – 2023 dataset variations divided by cluster. The smallest RMSE values are predominantly found in cluster 2, with 4 variables using the Adam optimizer and 2 other variables using the SGD optimizer. The variables with the smallest RMSE values in cluster 2 using the Adam optimizer are PM10, PM2.5, SO2, and NO2, while with the SGD optimizer are CO and O3. Upon observing Table 6, the difference in RMSE values between cluster 1 and cluster 2, whether using the Adam or SGD optimizer, is significant, with cluster 1 exhibiting relatively high RMSE values.

TABLE 5. RMSE SCORE FOR MULTIVARIATE ISPU 2021 – 2023 CLUSTER

Variable	Optimizer			
	Cluster 1		Cluster 2	
	Adam	SGD	Adam	SGD
PM10	14.46992	14.04687	**6.89939**	7.39796
PM2.5	25.64474	25.17816	**9.49449**	10.01863

Table 5 compares the multivariate LSTM models created for the ISPU 2021 – 2023 dataset per cluster. Multivariate here means that to predict PM10, the variable with high correlation values, in this study, PM2.5 is used, so the LSTM model input uses both PM10 and PM2.5 to predict PM10, and vice versa. The smallest RMSE values in this multivariate model for PM10 and PM2.5 variables are found in the Adam optimizer and also in cluster 2. Compared to the previous Table 4.6, which used variations per cluster on the ISPU 2021 – 2023 dataset, this multivariate LSTM model has smaller RMSE values compared to the univariate model per cluster for the same variables, PM10 and PM2.5

TABLE 6. RMSE SCORE FOR MULTIVARIATE ISPU 2021 – 2023 WHOLE DATASET

Variable	Optimizer	
	Adam	SGD
PM10	12.23176	**12.09990**
PM2.5	25.31911	**20.45937**
Temp	0.34370	**0.33801**
Humidity	**2.03769**	2.24793

Table 6 compares the RMSE values for the multivariate LSTM model using the entire ISPU 2021 – 2023 dataset and additional variables such as temperature and humidity. All variables from the ISPU parameters themselves obtained the smallest RMSE values when using the SGD optimizer, while the weather variables, namely temperature and humidity, obtained the smallest RMSE values with two different optimizers. The Adam optimizer was used for the temperature variable and the SGD optimizer for the humidity variable.

F. Result & Discussion

In general, the RMSE values generated by the LSTM modeling with inputs from the hybrid clustering results are smaller than those obtained using the entire dataset, despite treating all LSTM input datasets similarly with MinMaxScaler and splitting them into training-test data. The use of two different optimizers, Adam and SGD, also yields proportions or quantities that are not significantly different for models with the smallest RMSE values in the ISPU 2010 - 2023 and 2021 - 2023 datasets, both overall and per cluster. In the clustered dataset from ISPU 2010 - 2023, the smallest RMSE values in cluster 1 are achieved using the SGD optimizer, while for cluster 2, they are obtained with the Adam optimizer. Subsequently, the smallest RMSE values in the predictions using the hybrid cluster-acquired dataset in ISPU 2021 - 2023 are found in cluster 2, with the majority using the Adam optimizer. Cluster 2 in the ISPU 2021 - 2023 cluster results also exhibits smaller RMSE values compared to the smallest values across dataset variations. For the weather variables in the LSTM univariate model, the smallest RMSE values are observed in the 2021 - 2023 period. Regarding the multivariate model, the smallest RMSE values for the PM10 and PM2.5 variables are found in cluster 2 compared to using the entire dataset. Furthermore, the RMSE values for the multivariate model in temperature and humidity variables are not significantly different from those of the univariate model.

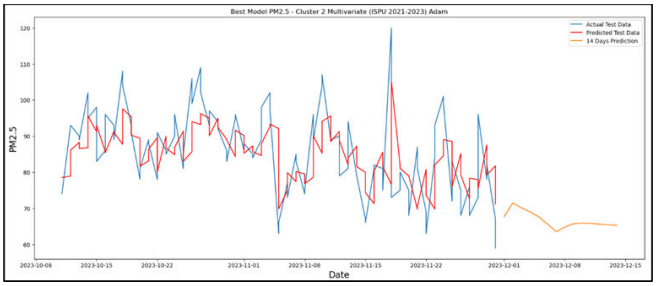


Fig. 5. Number of Category in ISPU 2021 – 2023 Clusters

Fig. 5 presents the visualization of prediction data from multivariate model using the Adam optimizer for the PM10 variable in the AQI dataset from 2021 to 2023. The resulting RMSE value is 6.89939. The prediction data pattern closely follows the actual data with minimal daily differences. This model incorporates both PM10 and PM2.5 variables to generate predictions for the PM10 column. The forecast for the next 14 days indicates a decrease in PM10 concentration based on the predictions of this multivariate model.

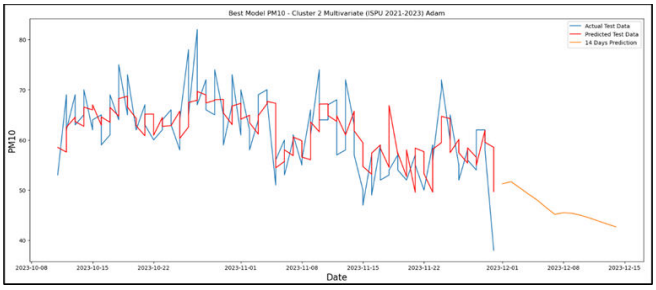


Fig. 6. Number of Category in ISPU 2021 – 2023 Clusters

Fig. 6 depicts the visualization of multivariate prediction data for the PM2.5 variable in the AQI dataset from 2021 to 2023 within cluster 2, employing the Adam optimizer. The RMSE value generated by this optimizer for this variable is 9.49449. This model integrates both PM10 and PM2.5 variables to forecast the PM2.5 variable. The prediction data pattern appears to closely mirror the actual data pattern, with minimal daily discrepancies. Moreover, the forecast for the next 14 days indicates a consistent trend following a decrease initially.

V. CONCLUSION

The research on the ISPU dataset of DKI Jakarta employed a hybrid model that combines clustering and predictive modeling stages using the CRISP-DM framework. The clustering stage utilized K-Means and K-Medoids algorithms to group data from air quality monitoring stations (SPKU)

within the ISPU dataset. This approach aimed to achieve balanced data proportions based on critical variables. Following clustering, Long Short-Term Memory (LSTM) models were used for predictive modeling, focusing on achieving the smallest root mean square error (RMSE) values. The study highlighted the significance of optimizer choice, revealing that certain data columns or variables performed better with specific optimizers. Notably, the LSTM model demonstrated superior performance in predicting air quality for the 2021-2023 ISPU dataset, with cluster 2 yielding the smallest RMSE values using the Adam optimizer.

The results underscored the importance of dataset size and optimizer selection in generating accurate predictive outcomes. For the ISPU dataset from 2010-2023, the smallest RMSE values in cluster 1 were achieved using the SGD optimizer, while cluster 2 performed best with the Adam optimizer. The study's findings suggest that the Adam optimizer is particularly effective for the 2021-2023 dataset's multivariate models, especially for predicting variables like PM10, PM2.5, and temperature. The research also highlighted the LSTM model's suitability for forecasting supporting data, such as weather variables. Future research suggestions include using ISPU datasets from different regions, incorporating additional supporting datasets, adjusting LSTM hyperparameters, and exploring various clustering algorithms to enhance the development of multivariate models and expand knowledge in data mining.

ACKNOWLEDGMENT

This research received support from the Institution of Research and Community Services at Universitas Multimedia Nusantara. We extend our appreciation to our colleagues at the Big Data Laboratory within the Information Systems Department at Universitas Multimedia Nusantara, whose valuable input and expertise greatly enriched the research.

REFERENCES

[1] "Sebanyak Ini Orang Mengeluh Soal Polusi Tiap Hari, Simak!" Accessed: Jan. 21, 2024. [Online]. Available: https://www.cnbcindonesia.com/news/20230822175926-4-465171/sebanyak-ini-orang-mengeluh-soal-polusi-tiap-hari-simak

[2] "Kualitas Udara Jakarta Tak Sebaik Kota Metropolitan Lainnya - Kompas.id." Accessed: Jan. 21, 2024. [Online]. Available: https://www.kompas.id/baca/riset/2023/06/22/kualitas-udara-jakarta-tak-sebaik-kota-metropolitan-lainnya

[3] M. Negara Lingkungan Hidup, "Keputusan Menteri Negara Lingkungan Hidup No. 45 Tahun 1997 Tentang : Indeks Standar Pencemar Udara".

[4] "Permen LHK No. 14 Tahun 2020." Accessed: Jan. 25, 2024. [Online]. Available: https://peraturan.bpk.go.id/Details/163466/permen-lhk-no-14-tahun-2020

[5] P. Govender and V. Sivakumar, "Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)," *Atmos Pollut Res*, vol. 11, no. 1, pp. 40–56, Jan. 2020, doi: 10.1016/J.APR.2019.09.009.

[6] W. Huang, T. Li, J. Liu, P. Xie, S. Du, and F. Teng, "An overview of air quality analysis by big data techniques: Monitoring, forecasting, and traceability," *Information Fusion*, vol. 75, pp. 28–40, Nov. 2021, doi: 10.1016/J.INFFUS.2021.03.010.

[7] K. Widjaja and R. S. Oetama, "K-Means Clustering Video Trending di Youtube Amerika Serikat," *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, vol. 11, no. 2, pp. 78–84, Dec. 2020, doi: 10.31937/SI.V11I2.1508.π

[8] Chirag, "Overview of Neural Network," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 531–534, Jun. 2022, doi: 10.48175/IJARSCT-4851.

[9] M. Sitorus, D. Fitron, and C. A. S. Wisesa, "Data Mining Implementasi Algoritma K-Means Menggunakan Aplikasi Orange dalam Clustering Pencemaran Udara di DKI Jakarta Tahun 2021," *Journal of Informatics and Advanced Computing (JIAC)* , vol. 3, no. 2, pp. 161–164, Nov. 2022, doi: 10.33633/tc.v14i4.992.

[10] N. M. Ariff, M. A. A. Bakar, and H. Y. Lim, "Prediction of PM10 Concentration in Malaysia Using K-Means Clustering and LSTM Hybrid Model," *Atmosphere 2023, Vol. 14, Page 853*, vol. 14, no. 5, p. 853, May 2023, doi: 10.3390/ATMOS14050853.

[11] A. Khumaidi, R. Raafi, I. Permana Solihin, and J. Rs Fatmawati, "Pengujian Algoritma Long Short Term Memory untuk Prediksi Kualitas Udara dan Suhu Kota Bandung," *Jurnal Telematika*, vol. 15, no. 1, pp. 13–18, Dec. 2020, doi: 10.61769/JURTEL.V15I1.340.

[12] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering," *Expert Syst Appl*, vol. 169, p. 114513, May 2021, doi: 10.1016/J.ESWA.2020.114513.

[13] N. G. Ali, S. D. Abed, F. A. J. Shaban, K. Tongkachok, S. Ray, and R. A. Jaleel, "Hybrid of K-Means and partitioning around medoids for predicting COVID-19 cases: Iraq case study," *Periodicals of Engineering and Natural Sciences*, vol. 9, no. 4, pp. 569–579, Oct. 2021, doi: 10.21533/PEN.V9I4.2382.

[14] M. A. Zen *et al.*, "Aplikasi Pengelompokan Data Runtun Waktu dengan Algoritma K-Medoids," *Inferensi*, vol. 6, no. 2, pp. 117–123, Sep. 2023, doi: 10.12962/J27213862.V6I2.15864.

[15] P. H. Gunawan, D. Munandar, and A. Z. Farabiba, "Long Short-Term Memory Approach for Predicting Air Temperature In Indonesia," *Jurnal Online Informatika*, vol. 5, no. 2, p. 161, Dec. 2020, doi: 10.15575/join.v5i2.551.

[16] A. Satyo and B. Karno, "Analisis Data Time Series Menggunakan LSTM (Long Short Term Memory) Dan ARIMA (Autocorrelation Integrated Moving Average) Dalam Bahasa Python.," *Ultima InfoSys : Jurnal Ilmu Sistem Informasi*, vol. 11, no. 1, pp. 1–7, Jul. 2020, doi: 10.31937/SI.V9I1.1223.

[17] J. Zhang, F. Chen, and Q. Shen, "Cluster-Based LSTM Network for Short-Term Passenger Flow Forecasting in Urban Rail Transit," *IEEE Access*, vol. 7, pp. 147653–147671, 2019, doi: 10.1109/ACCESS.2019.2941987.

[18] Y. Karyadi and H. Santoso, "Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 1, pp. 671–684, Mar. 2022, doi: 10.35957/JATISI.V9I1.1588.

[19] M. J. Hamid Mughal, "Data mining: Web data mining techniques, tools and algorithms: An overview," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 208–215, 2018, doi: 10.14569/IJACSA.2018.090630.

[20] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," *Procedia Comput Sci*, vol. 181, pp. 526–534, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.199.

[21] M. G. Sadewo, A. P. Windarto, and A. Wanto, "PENERAPAN ALGORITMA CLUSTERING DALAM MENGELOMPOKKAN BANYAKNYA

DESA/KELURAHAN MENURUT UPAYA ANTISIPASI/ MITIGASI BENCANA ALAM MENURUT PROVINSI DENGAN K-MEANS," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 2, no. 1, Oct. 2018, Accessed: May 05, 2024. [Online]. Available: https://ejurnal.stmik-budidarma.ac.id/index.php/komik/article/view/943

[22] R. Kesuma Dinata, N. Hasdyna, and N. Azizah, "Analisis K-Means Clustering pada Data Sepeda Motor," 2020.

[23] G. Dwilestari, I. Ali, R. P. Lunak, and S. T. Manajemen, "Analisis Clustering menggunakan K-Medoid pada Data Penduduk Miskin Indonesia," *JURSIMA (Jurnal Sistem Informasi dan Manajemen)*, vol. 9, no. 3, pp. 282–290, Sep. 2021, doi: 10.47024/JS.V9I3.302.

[24] A. A. D. Sulistyawati and M. Sadikin, "Penerapan Algoritma K-Medoids Untuk Menentukan Segmentasi Pelanggan," *SISTEMASI*, vol. 10, no. 3, p. 516, Sep. 2021, doi: 10.32520/stmsi.v10i3.1332.

[25] D. Ayu, I. C. Dewi, and K. Pramita, "Analisis Perbandingan Metode Elbow dan Sillhouette pada Algoritma Clustering K-Medoids dalam Pengelompokan Produksi Kerajinan Bali," 2019.

[26] K. Smagulova and A. P. James, "Overview of Long Short-Term Memory Neural Networks," *Modeling and Optimization in Science and Technologies*, vol. 14, pp. 139–153, 2019, doi: 10.1007/978-3-030-14524-8_11.

[27] C. B. Vennerød, A. Kjærran, and E. S. Bugge, "Long Short-term Memory RNN," May 2021, Accessed: Jan. 25, 2024. [Online]. Available: https://arxiv.org/abs/2105.06756v1

[28] M. F. Rizkilloh and S. Widiyanesti, "Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 25–31, Feb. 2022, doi: 10.29207/RESTI.V6I1.3630.

[29] F. N. Iman and D. Wulandari, "PREDIKSI HARGA SAHAM MENGGUNAKAN METODE LONG SHORT TERM MEMORY," *LOGIC : Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, no. 3, pp. 601–616, Apr. 2023, Accessed: Feb. 28, 2024. [Online]. Available: https://journal.mediapublikasi.id/index.php/logic/article/view/1855

[30] S. Sautomo, and H. Ferdinandus Pardede, "Prediksi Belanja Pemerintah Indonesia Menggunakan Long Short-Term Memory (LSTM)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 99–106, Feb. 2021, doi: 10.29207/RESTI.V5I1.2815.

[31] P. Saeipourdizaj, P. Sarbakhsh, and A. Gholampour, "Application of imputation methods for missing values of PM10 and O3 data: Interpolation, moving average and K-nearest neighbor methods," *Environmental Health Engineering And Management Journal*, vol. 8, no. 3, pp. 215–226, Jun. 2021, doi: 10.34172/EHEM.2021.25.

[32] S. A. Sanjaya and K. Surendro, "Spam Detection on Profile and Social Media Network using Principal Component Analysis (PCA) and K-means Clustering," *Int. J. Adv. Soft Compu. Appl*, vol. 11, no. 3, 2019.

turnitin Page 95 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Transfer Learning Approach for Rainfall Class Amount Prediction Using Uganda's Lake Victoria Basin Weather Dataset

Tumusiime Andrew Gahwera , Odongo Steven Eyobu*, Mugume Isaac
School of Computing and Informatics Technology
Makerere University
Kampala, Uganda
andrew.gahwera@mak.ac.ug, odongo.eyobu@mak.ac.ug*, isaac.mugume@mak.ac.ug

***Abstract*—Predicting short-term precipitation amounts is challenging especially due to meteorological data scarcity. Although deep Learning-based models have proven to be more effective in predicting precipitation their performance heavily depends on the size of the training dataset. This paper presents a Multi-station-based Transfer Learning approach with the aim of mitigating the data scarcity problem by transferring knowledge learned from multiple meteorological stations to a target station. In order to achieve this, a Multi-layer Perceptron, Convolutional Neural Networks, and Long-Short Term Memory systems were trained to predict rainfall class amounts on individual weather stations. From the experiments LSTM model outperformed the other-state-of-the-art models with an F1-score of 93% for sample stations of Jinja and Mwanza, and 95% for Musoma respectively. Consequently, the pre-trained LSTM model on each station were used as base models for Transfer Learning on the target station of Kisumu with limited data. The results show that the performance of the resulting transfer learning model improved by 3% for Jinja, Mwanza, and Musoma after model fine-tuning.**

***Index Terms*—precipitation, short-term forecasting, deep learning, multi-source transfer learning, meteorology**

I. Introduction

Limited access to meteorological information is a challenge in the prediction of rainfall amounts [1]. Additionally, in developing countries, challenges such as inadequate personnel at newly established weather stations, malfunctioning weather equipment, and power outages [2] make it difficult to gather sufficient amounts of meteorological data for prediction tasks. Despite these limitations, Numerical Weather Prediction (NWP) models [3] and Artificial intelligence (AI) based Machine learning (ML) [4] models have been successfully used to predict precipitation and other meteorological features like temperature, humidity and wind speed.

NWP models use mathematical equations and the experience of the modeling expert to make predictions [3]. However, these models fail to capture non-linear relationships in weather data [1], and are also computationally expensive [5]. Based on the existing studies [6], meteorological data is non-linear. Therefore, ML models are preferred over NWP models for precipitation prediction using non-linear data.

The increasing development of smart sensors and devices for massive data collection has accelerated the use of data-driven ML approaches [7]. These models are also flexible, more accurate, and adaptable to local conditions compared to NWP models [6].

Recently, Deep learning (DL) [8] models have been suggested for complex prediction scenarios involving non-linear datasets [6]. Subsequently, DL models were utilized for rainfall class amount prediction using the Lake Victoria basin weather dataset [9]. DL is a powerful machine learning technique that uses structured or unstructured data for classification using sophisticated decision-making process. The implementation of these models, however, require alot of historical training data [10], less of which the performance of the network is affected as its unable to capture important trends in the data.

Transfer Learning (TL) [10], [11] is a recent Deep learning technique which attempts to mitigate data-scarcity problems by transferring knowledge learned from large training dataset to a target dataset. This technique has been incorporated in several studies, such as in [11] where knowledge learned from one wind farm was fine-tuned to function on other wind farms. TL-based models showed better generalization in terms of root mean squared error (RMSE), mean absolute error (MAE), and standard deviation error (SDE) compared to existing techniques.

In [12], two transfer learning methods (fine-tuning and domain-adversarial neural network (DANN)) were used to improve daily precipitation quantity prediction by utilizing source domain data and target domain datasets. Based on the results, training the fine-tuning approach was quick and simple, and the MAE improved by 22.5%. However, the DANN method outperformed the fine-tuning method, towards better improvement of RMSE to 29.4%. In this study, we use a multi-station-based Transfer learning approach with the aim of mitigating the

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

data scarcity problem by transferring knowledge learned from multiple meteorological stations to a target station. In order to achieve this, a Multi-layer Perceptron, Convolutional Neural Networks, and Long-Short Term Memory systems were trained to predict rainfall class amounts on individual weather stations.

II. Related Works

The availability of sufficient amount and quality of training data is necessary for Deep learning methods to produce accurate results. In the absence of sufficient data, analytical methods face many difficulties, including but not limited to overfitting models and inappropriate generalization [10].

Transfer learning, a recent state-of-the-art DL technique, has shown promise in addressing data scarcity problems. TL uses relationships between two different but related datasets, tasks or models to transfer knowledge from the source domain to the target domain. There are various types of TL approaches but in this study, we use model-based transfer learning which uses a pre-trained model to transfer knowledge to the target dataset with limited data [13].

The performance of transfer learning is demonstrated in Xu et al. [14] work for flood forecasting in data sparse areas. Transfer learning framework based on Transformer (TL-Transformer) is used to accurately predict flooding in data-sparse basins (targets) by using models from data-rich basins (sources) without requiring extensive basin attributes at the target location. The Transformer model was preferred because of its ability to address the vanishing problem in LSTM architectures. Performance metrics like Nash Sutcliffe efficiency (NSE), RMSE, and bias were used to evaluate the models. The results show that TL-Transformer outperforms other state-of-the-art models in all target basin stations.

The work of Ambildhuke and Banik [15] used transfer learning technique to predict rainfall based on ground-based cloud images responsible for rains. The cloud images in the dataset were split into three classes labeled as no-rain to very low-rain, low to medium-rain, and medium to high rain with each class associated to which clouds are responsible for the appropriate rainfall. The CNN model was trained on the three respective classes mentioned above. Subsequently, the best pre-trained models including VGG16, Inception-V3, and XCeption with TL were used to improve model performance. From the experimental results, TL with the XCeption outperformed other models to predict rainfall using various image classes.

The general procedure of our TL approach, was to first train the model using the dataset with sufficient data, also known as the source dataset, then freeze some of the initial layers (further training will not change the weights for those layers when training is done), and lastly fine-tune the remaining layers using the dataset with limited data.

Transfer learning has been in existence for sometime, but its applicability in the domain of time series weather forecasting has been limited. In data sparse regions, like Uganda, it is important to build models that are generalizable. In the recent study [6], demonstrated that rainfall amounts in places which have comparable weather patterns can be predicted using machine learning regression models. Therefore, to further enhance generalizability of our prediction models in regions with inadequate meteorological data. This study utilized the Lake Victoria basin dataset [9] to demonstrate the efficacy of transfer learning in predicting hourly precipitation amounts based on rainfall classes (slight, moderate, heavy, and violent) rains.

III. Proposed Transfer Learning-based Precipitation Prediction Model

The deep learning model architectures that were used in this investigation include Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) and the Long Short-Term Memory (LSTM) models. Fig. 1 demonstrates the workflow of the pre-trained models and the transfer learning process in our study.

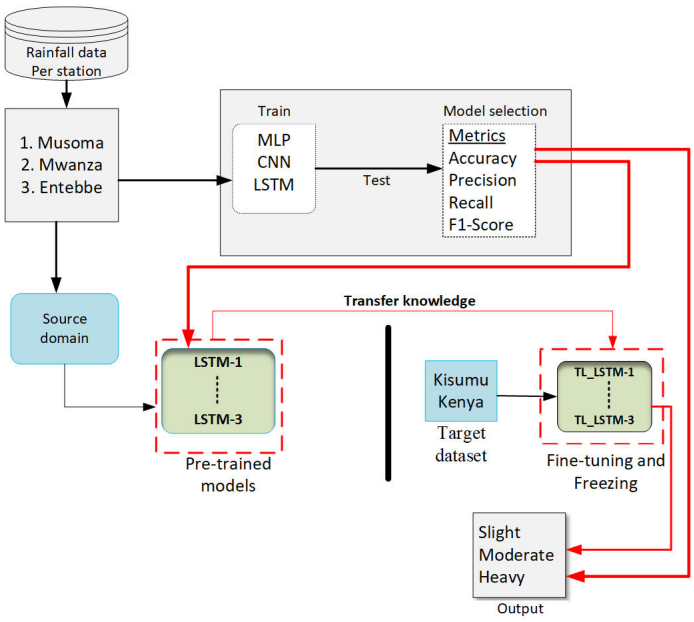


Fig. 1: The workflow of the proposed system

Our proposed approach makes use of rainfall data from 3 source stations to train the three deep learning models which are MLP, CNN and, LSTM. These models performances are then evaluated using defined metrics of accuracy, precision, recall and F1-score. The best pre-trained models from each station are then used to transfer knowledge to a target station to predict rainfall class amounts in the Lake Victoria basin.

Since we have considered 3 weather stations in this study, we obtain 3 models pre-trained on multiple source weather stations. Following that, each of these 3 models are fine-tuned on the training data of the target dataset. The different fine-tuning and freezing strategies are explored at this stage. The fine-tuned models are also subjected to model benchmarks, and

turnitin Page 97 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

are subsequently saved. The final procedure, we allow each pre-trained model of the 3 weather station to generate the prediction on the test data of the target dataset.

IV. Experimental Setup and Model Configurations

The study used Python [6] and several libraries for data preprocessing, modeling, evaluation, and visualization. The scikit-learn and Keras modules in Python, Numpy, pandas, imbalanced-learn, and matplotlib modules were utilized to process the time series weather data organization, data frame, and visualization. Due to space limitations, we present the hyperparameters of the tuned models below;

The optimum hyperparameters for MLP model are summarised in Table I.

TABLE I: The hyperparameters for MLP model.

Hyperparameters	Values
Number of hidden layers	(164,28)
Activation Function	ReLU
Output Layer activation function	Softmax
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Batch size	32
Epochs	50

The optimum hyperparameters for the CNN model are given in Table II.

TABLE II: The hyperparameters for CNN model.

Hyperparameters	Values
Number of Convolutional Filters	(32,64)
Kernel size	3
Pooling Layer	2
Flattening	Flatten()
Number of Neurons in Dense Layers	64
Activation Function	ReLU
Output Layer activation function	Softmax
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Batch size	32
Epochs	50

Table III summarises the optimum hyperparameters for the best resulting LSTM model.

V. Results and Discussions

In this section, we present the evaluation metrics used to measure the performance of the models. The classification models were evaluated using confusion matrix, accuracy, precision, recall, and F1-score. The matrix is shown in Table IV.

TABLE III: The hyperparameters for LSTM model.

Hyperparameters	Values
Number of LSTM Units	50
input shape	(X_train_smote.shape[1], 1)
Return Sequences	return_sequences=True
Activation Function	ReLU
Output Layer activation function	Softmax
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Batch size	32
Epochs	50

TABLE IV: Confusion Matrix.

		Predicted values	
		Positive	Negative
Actual values	Positive	TP	FN
	Negative	FP	TN

A. Evaluation metrics

The list of evaluation metrics used for evaluating trained classifiers are stated in Eqs (1)–(4).

$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions} \quad (1)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3)$$

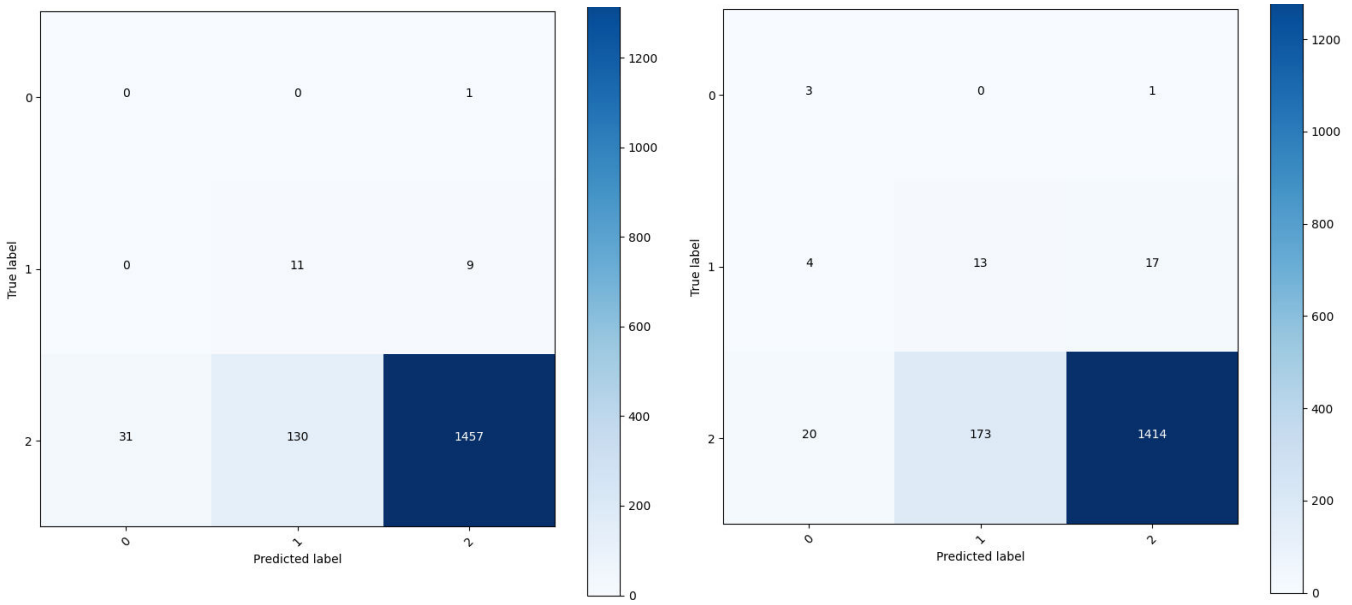
$$F1 - Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (4)$$

B. Performance comparison with state-of-the-art models

The performance of the pre-trained LSTM models were compared with two (2) state-of-the-art deep learning architectures from literature to validate their effectiveness. The 3 years of hourly observational weather data from three source stations are presented for this purpose. The train-test split of 80% and 20% was used across all the 3 weather stations to ensure fairness in comparison. Table V shows the performance of the best LSTM models with the other model architectures used in this study. In Table V, the LSTM models outperformed the other state-of-the-art classification models in predicting rainfall class amounts around the Lake Victoria basin.

Fig. 2 shows the confusion matrix of the best performing LSTM models for Mwanza and Bukoba station sample showing precision, recall, and F1-score for rainfall classes

turnitin Page 98 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772



(a) (b)

Fig. 2: The confusion matrix results of sample station datasets. (a) LSTM-Model Mwanza. (b) LSTM-Model Bukoba.

slight, moderate, and heavy rains. The corresponding values for each class are slight = 2, moderate = 1, and heavy = 0 [9]. Also, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data to address class imbalances [16]. The results of Fig. 2 (a) and 2 (b) show the individual class accuracy predictions of the two LSTM models per station dataset. In particular, Fig. 2 (a) indicate 1,457 cases were correctly predicted by LSTM algorithm as slight rainfall which was actually true. 130 counts of cases of slight rains were incorrectly predicted as moderate. 31 cases of slight rainfall were incorrectly predicted as heavy rainfall. 9 cases of moderate were incorrectly predicted by LSTM as slight rainfall. 11 cases were correctly predicted as moderate rainfall. No cases of of moderate rainfall were incorrectly predicted as heavy rainfall. 1 case of heavy rainfall was incorrectly predicted as slight rainfall. Furthermore, no case of heavy rainfall was incorrectly predicted as heavy rainfall. Finally, no heavy rainfall was observed in this station. Fig. 2 (b) shows 1,414 cases of slight rainfall, 13 cases of moderate rainfall, and 3 cases of heavy were correctly predicted by the LSTM model for this station. Generally, the LSTM model demonstrated acceptable class distribution for all stations confusion matrices; some are not shown because of space limitations.

1) Explainability of the LSTM Model: The study evaluated the predictive capabilities of LSTM models across various station datasets, employing SHapley Additive exPlanation (SHAP) to interpret the model outputs [17]. The SHAP analysis aimed to quantify each input's contribution to the prediction. Fig. 4 illustrates the SHAP results for the Bukoba weather station.

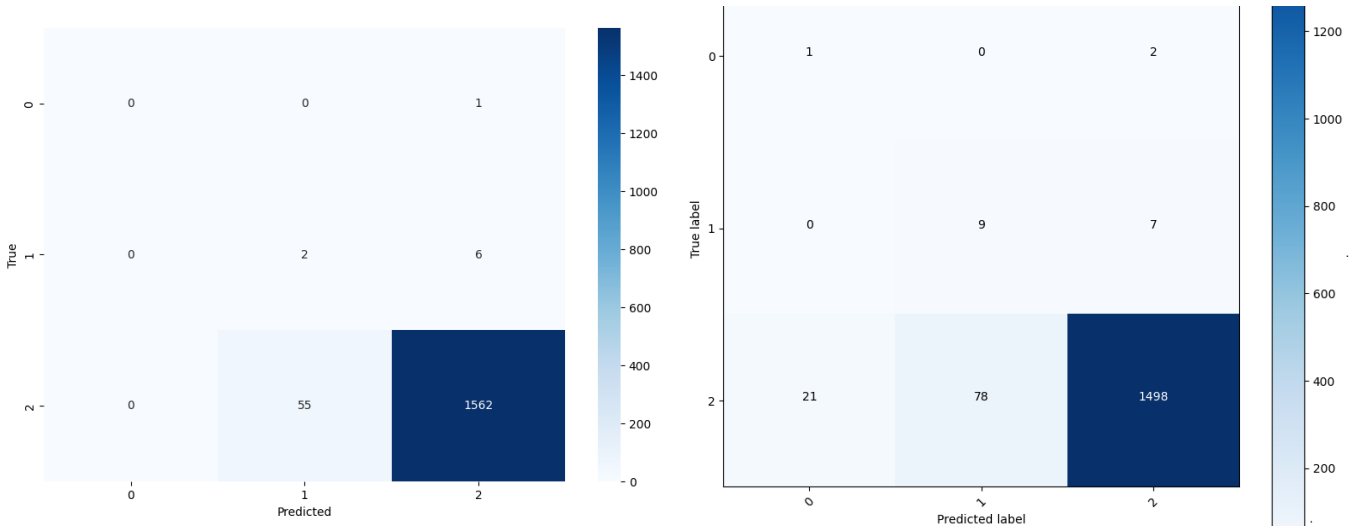
TABLE V: Performance comparison of the applied models.

Station	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Musoma	MLP	0.90	0.98	0.90	0.94
	CNN	0.89	0.99	0.89	0.93
	LSTM	0.93	0.98	0.93	**0.95**
Mwanza	MLP	0.83	0.98	0.83	0.90
	CNN	0.85	0.98	0.85	0.91
	LSTM	0.90	0.98	0.90	**0.93**
Jinja	MLP	0.84	0.98	0.84	0.90
	CNN	0.83	0.98	0.83	0.89
	LSTM	0.88	0.98	0.88	**0.93**

In Fig. 4, the red dots indicate features with a significant impact on the model's performance. The analysis revealed that the visibility (Viskm) feature did not positively contribute to the LSTM model's performance for the Bukoba station, a finding consistent across all other weather stations analyzed. In contrast, the remaining features exhibited varying degrees of positive influence on the LSTM models' outputs from the utilized weather station datasets. This highlights the importance of identifying key features in rainfall prediction models, particularly for the Lake Victoria basin.

C. Transfer Learning results

In this subsection the performance of the pre-trained LSTM models for the source stations of Musoma, Mwanza, and Jinja weather stations are used to transfer knowledge to the



(a) (b)

Fig. 3: The confusion matrix results of sample station Musoma. (a) TL-LSTM: Fine-tuning. (b) TL-LSTM: Freezing.

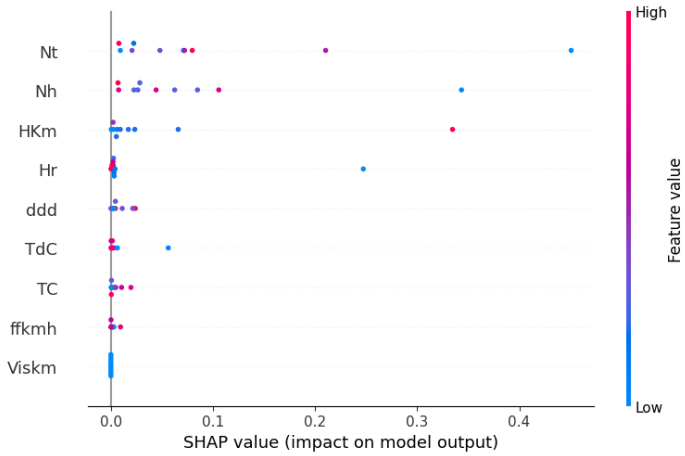


Fig. 4: Feature importance of the best performing LSTM model for Bukoba station sample dataset.

target station Kisumu in the case of data scarcity scenario as presented in Table VI and VII. In this case, 7 months (40%) of 3 years observational weather data is used for the target dataset, Kisumu. The pre-trained LSTM models were trained independently on each of the 3 source stations, resulting to 3 pre-trained models. Three years (36 months) of data was considered for the source stations.

Subsequently, each of the pre-trained models are used to transfer knowledge to the target dataset on individual basis. Transferring knowledge involves freezing certain layers of the pre-trained model architecture and allowing the weights of the remaining layers to be changed. Two strategies for fine-tuning are tested. In the first strategy, all layers were kept fine-tunable on a new dataset. In the second case, trainable layers were frozen while the LSTM layer and dense layer were allowed to be fine-tuned.

1) Fine-tuning strategy 1: Table VI shows the performance values obtained for predictions on the target station (Kisumu). The table shows the results of knowledge transfer from models pre-trained on each source station. From table VI we can clearly see that TL outperformed the pre-trained LSTM models for the different weather stations as shown in Table V. Fig. 3 (a) shows the confusion matrix for fine-tuning strategy 1.

TABLE VI: Performance of single-source TL, fine-tuning strategy 1.

Metric	Musoma	Mwanza	Jinja
Accuracy (%)	0.96	0.96	0.98
Precision (%)	0.99	0.93	0.95
Recall (%)	0.96	0.91	0.93
F1-score (%)	**0.98**	**0.96**	**0.96**

2) Freezing strategy 2: Table VII shows the performance values of freezing strategy 2 obtained on the target station (Kisumu) for the three weather stations. The results show the superior performance of knowledge transfer over the single source pre-trained models shown in Table in V. However, the fine-tuning strategy 1 performs better than the freezing strategy 2 by 2% for Musoma, 1% for both Mwanza and Jinja station datasets. This is attributed to the flexibility of the pre-trained model to adapt to the specific features of the new dataset which helps in achieving higher accuracy and better generalization

[18]. The confusion matrix of LSTM model with TL strategy 2 for Musoma station sample is shown in Fig. 3 (b). But generally, TL approach performs better than the pre-trained base models from this particular weather station.

TABLE VII: Performance of single-source TL, Freezing strategy 2.

Metric	Musoma	Mwanza	Jinja
Accuracy (%)	0.93	0.95	0.97
Precision (%)	0.98	0.93	0.94
Recall (%)	0.93	0.92	0.95
F1-score (%)	**0.96**	**0.95**	**0.95**

VI. CONCLUSION AND FUTURE WORK

This study explored the approach of multi-source station transfer learning to mitigate the data scarcity problem associated with deep-learning systems to improve prediction performance. The study used meteorological features that influence rainfall class amount prediction in the Lake Victoria basin.

Deep Learning models including MLP, CNN, and LSTM were trained and tested on multiple weather station datasets. The results of the models were compared and the LSTM pre-trained model was best performing model across the 3 weather stations in the Lake Victoria basin. Consequently, the LSTM pre-trained models were used to transfer knowledge on each weather station using the the target dataset, Kisumu. The results proved that TL is an effective method for predicting rainfall class amounts in areas with limited meteorological data.

Future work will apply ensemble of the TL-LSTM base models from the multiple weather stations to a different target dataset. We strongly believe that ensemble models will further improve the accuracy of individual TL models across the various weather stations. The methodology of knowledge transfer learned from multiple sources to a single target dataset, has been limited in the domain of precipitation prediction. Thus, our work can be used to further improve weather forecasting research.

ACKNOWLEDGMENT

This work was supported by Makerere University, Research and Innovation Fund (RIF) under the Government of Uganda.

REFERENCES

[1] V. S. Monego, J. A. Anochi, and H. F. de Campos Velho, "South america seasonal precipitation prediction by gradient-boosting machine-learning approach," *Atmosphere*, vol. 13, no. 2, p. 243, 2022.

[2] A. Meque, S. Gamedze, T. Moitlhobogi, P. Booneeady, S. Samuel, and L. Mpalang, "Numerical weather prediction and climate modelling: Challenges and opportunities for improving climate services delivery in southern africa," *Climate Services*, vol. 23, p. 100243, 2021.

[3] K. Yonekura, H. Hattori, and T. Suzuki, "Short-term local weather forecast using dense weather station by deep neural network," pp. 1683–1690, 2018.

[4] J. Gu, S. Liu, Z. Zhou, S. R. Chalov, and Q. Zhuang, "A stacking ensemble learning model for monthly rainfall prediction in the taihu basin, china," *Water*, vol. 14, no. 3, p. 492, 2022.

[5] C. Huntingford, E. S. Jeffers, M. B. Bonsall, H. M. Christensen, T. Lees, and H. Yang, "Machine learning and artificial intelligence to aid climate change research and preparedness," *Environmental Research Letters*, vol. 14, no. 12, p. 124007, 2019.

[6] T. A. Gahwera, O. S. Eyobu, and M. Isaac, "Analysis of machine learning algorithms for prediction of short-term rainfall amounts using uganda's lake victoria basin weather dataset." *IEEE Access*, 2024.

[7] D. Kim, Y. Lee, K. Chin, P. J. Mago, H. Cho, and J. Zhang, "Implementation of a long short-term memory transfer learning (lstm-tl)-based data-driven model for building energy demand forecasting," *Sustainability*, vol. 15, no. 3, p. 2340, 2023.

[8] E. Hernández, V. Sanchez-Anguix, V. Julian, J. Palanca, and N. Duque, "Rainfall prediction: A deep learning approach," in *Hybrid Artificial Intelligent Systems: 11th International Conference, HAIS 2016, Seville, Spain, April 18-20, 2016, Proceedings 11*. Springer, 2016, pp. 151–162.

[9] A. G. Tumusiime, O. S. Eyobu, I. Mugume, and T. J. Oyana, "A weather features dataset for prediction of short-term rainfall quantities in uganda," *Data in Brief*, vol. 50, p. 109613, 2023.

[10] A. Dhole, I. Ambekar, G. Gunjan, and S. Sonawani, "An ensemble approach to multi-source transfer learning for air quality prediction," in *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2021, pp. 70–77.

[11] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," *Applied Soft Computing*, vol. 58, pp. 742–755, 2017.

[12] Z. Liu, Q. Yang, J. Shao, G. Wang, H. Liu, X. Tang, Y. Xue, and L. Bai, "Improving daily precipitation estimation in the data scarce area by merging rain gauge and trmm data with a transfer learning framework," *Journal of Hydrology*, vol. 613, p. 128455, 2022.

[13] P. Wang, Z. Chen, X. Deng, J.-S. Wang, R. Tang, H. Li, S. Hong, and Z. Wu, "The short-time prediction of thermospheric mass density based on ensemble-transfer learning," *Space Weather*, vol. 21, no. 10, p. e2023SW003576, 2023.

[14] Y. Xu, K. Lin, C. Hu, S. Wang, Q. Wu, L. Zhang, and G. Ran, "Deep transfer learning based on transformer for flood forecasting in data-sparse basins," *Journal of Hydrology*, vol. 625, p. 129956, 2023.

[15] G. M. Ambildhuke and B. G. Banik, "Transfer learning approach-an efficient method to predict rainfall based on ground-based cloud images." *Ingénierie des Systèmes d'Information*, vol. 26, no. 4, 2021.

[16] S. Kanani, S. Patel, R. K. Gupta, A. Jain, and J. C.-W. Lin, "An ai-enabled ensemble method for rainfall forecasting using long-short term memory," 2023.

[17] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.

[18] R. Oruche, L. Egede, T. Baker, and F. O'Donncha, "Transfer learning to improve streamflow forecasts in data sparse regions," *arXiv preprint arXiv:2112.03088*, 2021.

turnitin Page 101 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Forecasting the NBA's Most Valuable Player: A Regression Analysis Approach

Arnando Harlianto
Information Systems Study
Universitas Multimedia Nusantara
Tangerang, Indonesia

Johan Setiawan
Information Systems Study
Universitas Multimedia Nusantara
Tangerang, Indonesia

Abstract— Basketball is immensely popular in the United States, with the National Basketball Association (NBA) recognized as the leading professional league worldwide. Each season, the NBA awards the Most Valuable Player (MVP) title to the player considered the best based on performance statistics and a fair voting process. Predicting the MVP often sparks debates among NBA fans, with various media outlets offering their forecasts. This study aims to accurately predict the MVP by analyzing statistical data from NBA players over a specific period. We employ three different regression algorithms—Decision Tree, Linear Regression, and Support Vector Regression—to compare their effectiveness in MVP prediction. The research findings will highlight the potential MVP winner based on their performance statistics, presented in a simplified dashboard format for easy comprehension..

Keywords: Decision Tree, Linear Regression, Most Valueable Player, National Basketball Association, Support Vector Regression.

I. INTRODUCTION

Basketball is one of the most popular sports in the United States. The organizations and leagues involved in basketball in the United States are very active and continuously hold various matches every season. Matches are held every season in each league. The leagues vary based on age and skill level, ranging from NCAA, which is the league for universities, to the NBA, which is the highest professional league in the United States. In every professional league in the United States, such as the NBA, award programs are held for specific players each season. These awards are given to players with the best statistics of the season. The purpose of these awards is to signify to the players that they have the best statistical value (most valuable) of the season. The Most Valuable Player (MVP) award in the NBA provides several benefits to the winners. Players with the MVP title can be dubbed the best player in the NBA at that time. This undoubtedly affects the contracts the players receive and also enhances their fame. The MVP award itself is determined through a voting system [1].

In addition to media, there are also scholarly journal articles that predict basketball performance using valid research methods. Research in this area is conducted using spatial tracking data. By utilizing spatial tracking data, researchers can conduct detailed investigations into strategies, player performance, team performance, and more, accurately and in detail. The technology employed here also enables comparisons and data matching of players to analyze their performance during both practice sessions and actual games [2].

The National Basketball Association (NBA) is one of the largest basketball organizations in the world. Every season, the NBA always holds various series of matches, starting from pre-season games, regular-season games, to playoffs. Each year, the NBA will produce one winner (NBA Champions), and at the end of the season, the NBA will announce various other awards such as MVP (Most Valuable Player), DPOY (Defensive Player of The Year), ROTY (Rookie of The Year), and others. However, for awards, MVP is the highest award for the best players, such as Michael Jordan, LeBron James, Kobe Bryant, who are three of the best players in the NBA who have each received the MVP title with clear statistics or can be said to be clearly the best players of the season. [3]. Fig. 1 shows the Average predicted MVP share.

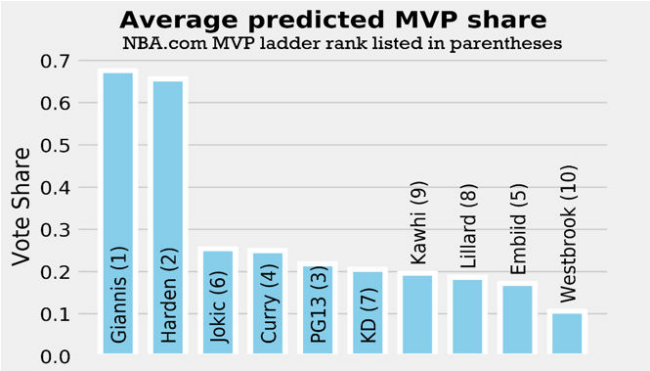


Fig. 1 Average predicted MVP Share
Source: dribble.com

The National Basketball Association (NBA) operates on an annual basis, commonly referred to as each season. At the conclusion of every NBA season, distinguished titles are awarded to both teams and players. Teams vie for the prestigious title of NBA Champions, while individual players aspire to earn the MVP (Most Valuable Player) award.

The MVP title is determined through a voting process involving 100 independent media members who are not affiliated with any NBA team. Each voter nominates five candidates for the MVP, ranking them from first to fifth. Points are allocated as follows: 10 points for the first-place vote, 7 for the second, 5 for the third, 3 for the fourth, and 1 for the fifth. The player with the highest total points is awarded the MVP title. Utilizing appropriate and relevant methods, predictions can yield accurate results to forecast the MVP recipient each season [4]. This research will interconnect various variables and employ multiple algorithms to compare their predictive accuracy. Data visualization will also be conducted to enhance the understanding of the data and present results with more effective and objective visuals.

This study differentiates itself from previous research by employing distinct datasets and algorithms. It uses NBA player data to predict the MVP for the 2022-2023 season. Unlike previous studies, which utilized Neural and Regression Models to predict sports outcomes, this research employs Regression Models with three different algorithms: Decision Tree, Linear Regression, and Support Vector Regression. The

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

variable used to gauge players' likelihood of receiving the MVP title is **Fantasy points**, chosen because they represent a comprehensive statistical value derived from each player's performance across various categories.

II. LITERATURE REVIEW

A. Machine Learning

Machine Learning is a technology designed to learn and expand its knowledge autonomously, without user direction. It has rapidly advanced and is now integral to developing software features such as voice recognition, natural language processing, and robot control [5] The term "Machine Learning" was coined by mathematicians Thomas Bayes, Adrein Marie Legendre, and Andrey Markov in the 1920s, laying the conceptual foundation for the field. Machine Learning enhances machines' ability to handle data more efficiently [6].

There are two main types of Machine Learning:

Supervised Learning: This type involves training data with input-output pairs. It is used for classifying unseen data into predefined categories and making predictions [7]

Unsupervised Learning: This type processes raw or unlabeled data to identify patterns and trends, grouping data into specific clusters [7]

B. Decision Tree

Decision Tree is a supervised regression algorithm that uses a set of rules to form a tree-like structure, mirroring human decision-making processes, making it easy to understand. It can handle both discrete and continuous data inputs. Decision Trees provide a graphical representation of the decision-making process under specific conditions. They are commonly used to evaluate whether decisions based on given data are optimal. This method has broad applications across various fields, including manufacturing, social issues, business settings, and service industries such as banking, libraries, hospitals, and pharmaceuticals, benefiting both small and large enterprises [8].

C. Linear Regression

Linear Regression is a widely used supervised learning algorithm in data science. It applies regression principles to create predictive models for target variables based on other independent variables [9]. This algorithm is commonly employed to identify correlations between variables and generate accurate predictions. The primary goal is to predict a dependent variable (Y) using given independent variables (X), resulting in a linear relationship model between input (X) and output (Y). Linear Regression is favored for its simplicity, ability to provide accurate predictions across various case studies, and ease of interpretation, making it one of the most popular algorithms for regression analysis.

D. Support Vector Regression

Support Vector Regression (SVR) is a machine learning algorithm used to model the relationship between a dependent variable and one or more independent variables [10], [11]. It aims to predict the dependent variable's values based on given independent variables. As a variant of Support Vector Machine (SVM) [12], SVR is designed for regression tasks to predict continuous numerical values, making it suitable for cases like stock price forecasting. The goal of SVR is to find a regression model with a margin around the predicted values, balancing data fit and avoiding overfitting. This approach ensures accurate predictions. SVR is commonly implemented using Python's scikit-learn library, which provides robust tools for machine learning tasks.

E. Related Works

In previous studies [11], Linear Regression has demonstrated outstanding accuracy, highlighting its efficacy in modeling relationships between variables and generating precise forecasts. Support Vector Regression (SVR) also delivered notable results, positioning itself as a strong contender alongside Decision Tree and Linear Regression.

Other research [13] has shown SVR's superior accuracy compared to its counterparts, emphasizing the versatility of regression algorithms in diverse research contexts. These findings highlight the importance of considering multiple algorithms to ensure a comprehensive analysis and informed decision-making. Furthermore, they underscore SVR's potential as a powerful tool for extracting valuable insights from complex datasets, paving the way for advancements in predictive modeling and data-driven strategies.

III. METHODOLOGIES

CRISP-DM is used as a frame work for this research. The Cross Industry Standard Process for Data Mining (CRISP-DM) is a data mining framework that serves as a basis for conducting data science or data mining processes. It consists of six sequential stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. These six stages must be understood and executed sequentially to yield the best and most accurate results in a data mining project [14][15].

A. Business Understanding

This is the first stage of CRISP-DM. It involves understanding the business and the use of data mining within it. In this research, data mining is used to assist media in predicting which player will receive the MVP title in a specific season of the NBA.

B. Data Understanding

Data Understanding is the stage of understanding the data. In this stage, data is collected and its quality is examined. Afterward, data checking will be performed starting from the variables and fields within the statistical data. The statistical data obtained and used in this research are valid and reliable as they are collected from the official NBA source itself.

C. Data Preparation

Data Preparation is the stage where data is prepared to be used in a mature and accurate manner. In this research, NBA statistics data will be prepared in detail, starting from the data cleansing process to remove various factors that could destabilize the data's stability, and filtering the data where unnecessary fields or variables can be eliminated. This stage is crucial in the analysis process because it ensures the quality of the data and prepares it for use in a study. Not only that, but data readiness will also affect the level of accuracy produced in the future.

D. Modeling

Modeling is the stage of selecting algorithms and forming models in a study. In this stage, Machine Learning models will be created with predetermined algorithms. In this research, Machine Learning models will be created and formed using three different algorithms: Decision Tree, Linear Regression, and Support Vector Regression.

E. Evaluation

Evaluation is the stage of evaluating the models formed in the Modeling stage. In this stage, the three models will be evaluated in detail. In this research, the three models will be tested and compared with each other. This is done to see if the formed algorithmic models are suitable or not, and also to see which algorithmic model is most suitable and has the best accuracy results.

F. Deployment

Deployment is the final stage of CRISP-DM. In this stage, the most suitable or best model will be applied to the data. Not only that, but this stage also includes monitoring the models used. This is done to reduce the risk of problems occurring and to anticipate problems that will arise during the implementation process. In this research, the models that have been created and evaluated will be applied to NBA statistical data and continuous monitoring will be carried out to oversee the process in order to produce high and accurate accuracy results.

IV. ANALYSIS AND RESULTS

A. Business Understanding Stage

This study aims to predict the Most Valuable Player (MVP) in the National Basketball Association (NBA) using regression methods. By leveraging player statistics from active NBA players, we seek to create an accurate predictive model. The regression algorithms employed are Decision Tree, Linear Regression, and Support Vector Regression. The goal is to develop a reliable model that can predict the MVP before the season ends, providing valuable insights with high accuracy.

B. Data Understanding Stage

The data for this study comprises statistical information from active NBA players, sourced from the official NBA website, NBA.com. The dataset includes player statistics from the 2022/2023 season, encompassing 539 rows and 28 columns. These rows represent all the players in the NBA.

The primary metric used for prediction is fantasy points, calculated based on various individual statistics such as points, rebounds, assists, steals, blocks, and turnovers. The fantasy points are assigned as follows: 3 points for every 3-point shot, 2 points for every 2-point shot, 1 point for free throws, 1.2 points for rebounds, 1.5 points for assists, 2 points for steals and blocks, and a deduction of 1 point for turnovers. Analyzing fantasy points provides a comprehensive view of player performance and consistency throughout the season.

The hypothesis is that players with high fantasy points are likely to have a significant and consistent impact, making them strong candidates for the MVP award. The detailed attributes of the dataset are listed in Table 1.

Table 1 Data Attributes

No.	Attribute	Type	Description
1	player	string	Name of the NBA player
2	team	string	Name of the team
3	age	int64	Age of the player
4	games_played	int64	Number of games played
5	wins	int64	Number of wins
6	loses	int64	Number of losses
7	minutes_played	float64	Average minutes played per game
8	points	float64	Average points scored per game
9	field_goals_made	float64	Field goals made per game
10	field_goals_attempted	float64	Field goals attempted per game
11	field_goals_percentage	float64	Field goal percentage
12	3_point_made	float64	Three-point shots made per game
13	3_point_attempted	float64	Three-point shots attempted per game
14	3_point_percentage	float64	Three-point shot percentage
15	free_throws_made	float64	Free throws made per game
16	free_throws_attempted	float64	Free throws attempted per game
17	free_throws_percentage	float64	Free throw percentage
18	offensive_rebounds	float64	Offensive rebounds per game
19	defensive_rebounds	float64	Defensive rebounds per game
20	rebounds	float64	Average rebounds per game
21	assists	float64	Average assists per game
22	turnovers	float64	Average turnovers per game
23	steals	float64	Average steals per game
24	blocks	float64	Average blocks per game
25	personal_fouls	float64	Average personal fouls per game
26	fantasy_points	float64	Fantasy points based on game statistics
27	double_doubles	int64	Number of double-doubles (10+ in two statistical categories in a game)
28	triple_doubles	int64	Number of triple-doubles (10+ in three statistical categories in a game)

This study does not involve interviews for data collection. Instead, we focus on analyzing publicly available statistical data to identify factors influencing fantasy points in the NBA. This approach ensures objectivity and reliability, enabling a comprehensive and in-depth analysis without the need for subjective input. By utilizing existing data, we aim to provide robust insights into the determinants of fantasy points and their correlation with MVP candidacy

C. Data Preparation Stage

Data preparation is a critical phase where raw data is transformed into a refined format ready for analysis. This involves several steps including importing necessary libraries, labeling data, adding columns for analysis, and checking for and cleaning any null values. These steps ensure that the data is more accessible, understandable, and ready for in-depth analysis.

The data used in this study was sourced from the official NBA website, NBA.com, through web scraping using a Google extension tool. The extracted data was compiled into a dataset in Excel or .csv format.

NBA Player Statistics (2022/2023)

The dataset includes comprehensive statistics for NBA players from the 2022/2023 season, sorted by fantasy points

earned during the regular season. This limitation ensures the data is consistent and relevant.

Creating a Sitemap for Data Scraping

A sitemap was created to select the specific data to be scraped. This involved naming the sitemap, inputting URLs, selecting data, and specifying the rows and columns to be extracted.

Importing Libraries

The first step in the data preparation process was importing all necessary libraries. These libraries facilitate various tasks from data reading and modification to model creation and result visualization.

Sorting Data by Fantasy Points

Data from the 2021/2022 and 2022/2023 seasons were sorted by fantasy points. This sorting is crucial as the primary variable to be predicted is fantasy points, aiding in a more streamlined analysis process.

Checking Data Variable Types

Each variable in the dataset was checked for its data type to ensure a clear understanding of the data structure. This step helps in preparing the data for further analysis.

Checking for Null Values

The dataset was thoroughly checked for null values, and it was confirmed that there were no null values present. This indicates that the data is clean and ready for analysis without the need for additional data cleaning processes.

Splitting and Training Test Data

The data was split into variables x and y. Variable x included attributes such as age, games played, wins, losses, minutes_played, points, field_goals_made, field_goals _attempted, field_goal_percentage, three-point_shots_made, three-point_shots_attempted, three-point_percentage, free _throws_made, free_throws_attempted, free_throw_ percentage, offensive_rebounds, defensive_rebounds, total_ rebounds, assists, turnovers, steals, blocks, personal fouls, double-doubles, and triple-doubles. Variable y comprised the fantasy points.

The dataset was then divided into a training set (80%) and a testing set (20%). The training set was used to build the model, while the testing set served as unseen data to evaluate the model's performance. This ratio ensures that the model can generalize well to new data.

Monitoring and Updating the Model

As training data evolves, it is crucial to assess the significance of these changes. Minor changes may not impact the model's performance significantly, but conceptual changes or new data availability necessitate model retraining or updates to maintain accuracy and relevance. Regular performance monitoring and evaluation against the latest data are essential to maintain the quality of the machine learning model.

D. Modeling Stage

The modeling stage is crucial for understanding and solving complex problems. It allows for the exploration of hypotheses, validation of theories, and gaining detailed insights into the topic. This research employs regression as a key tool to build relationships between relevant variables. Specifically, three regression algorithms are used: Decision Tree, Linear Regression, and Support Vector Regression.

The objective is to predict the fantasy points of NBA players based on their season statistics. Fantasy points are chosen due to their correlation with a player's likelihood of being named the Most Valuable Player (MVP). However, fantasy points alone do not determine the MVP, as the NBA uses additional processes, such as the MVP Race, which involves subjective voting by trusted analysts and media, known as MVP Shares. This process is inherently subjective, as the voting is conducted by trusted analysts and observers selected by the NBA for the voting process [16]

Factors Influencing MVP Selection through Fantasy Points:

Statistical Analysis: Fantasy points are accumulated from key statistics like points, rebounds, assists, and steals. These components are critical in evaluating a player's performance and their contribution to the team.

Fan and Media Influence: The popularity of fantasy basketball among fans and analysts means that players with high fantasy points often receive more attention. This media exposure can influence public perception and MVP voting [16].

Significant Contribution: High fantasy points indicate significant contributions to the team, increasing a player's chances of being nominated for MVP.

While fantasy points play a significant role in the MVP selection process, it is essential to remember that the final decision involves various subjective factors, including a player's impact both on and off the court. For instance, in the 2018/2019 season, James Harden was named MVP, supported by his impressive fantasy point statistics—averaging 64.53 fantasy points per game and 58.7 during regular games. His high fantasy points underscored his dominant performance at that season, reinforcing his MVP candidacy.

Table 2 Comparison of Actual Data and Predictions

Player	Actual Data	Prediction		
		DT	LR	SVR
Luka Doncic	56.8	56.2	56.7	36.8
Joel Embiid	56.2	56.2	56.2	36.6
Nikola Jokic	55.7	40.1	55.4	33.7
Giannis Antetokounmpo	54.8	54.8	54.7	35.3
Anthony Davis	52.0	52.0	51.8	34.8
Shai Gilgeous-Alexander	50.4	50.4	50.4	31.6
LeBron James	50.3	41.1	50.3	33.3
Jayson Tatum	49.9	49.9	49.9	35.4
Damian Lillard	49.1	49.1	49.1	33.3
Kevin Durant	47.7	41.1	47.5	29.6
Stephen Curry	46.8	44.8	46.8	31.5
James Harden	46.2	31.2	46.0	32.6
Ja Morant	46.0	46.0	46.1	31.8
Domantas Sabonis	45.8	45.8	45.7	33.8
Kyrie Irving	44.8	44.8	45.0	30.8
LaMelo Ball	44.8	44.8	44.6	27.9
Trae Young	44.6	44.6	44.6	34.6
Tyrese Haliburton	44.5	44.5	44.2	31.4
Pascal Siakan	44.3	40.1	44.0	34.1
Julius Randle	43.2	43.2	43.1	35.2

DT= Decision Tree, LR=Linear Regression, SVR=Support Vector Regression

Table 2 provides a summary of the actual data compared with predictions made by the Decision Tree, Linear Regression, and Support Vector Regression models.

E. Evaluation Stage

The evaluation stage is conducted after the development of the three regression models: Decision Tree, Linear Regression, and Support Vector Regression. This critical step involves assessing the performance of each model using a pre-established validation dataset.

Validation Dataset: A separate validation dataset, distinct from the training set, is used to ensure an unbiased evaluation of model performance. This dataset includes player statistics from the 2022/2023 NBA season, which the models have not previously encountered.

Performance Metrics: The models are evaluated based on key performance metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R2 score. These metrics provide a comprehensive view of each model's accuracy and predictive power.

Results: The performance of each model on the validation dataset is summarized below:

Table 3 – Performance Comparison

Algorithm	R2 Score	MSE Score	RMSE Score
Decision Tree	92.55%	13.27	2.64
Linear Regression	99.99%	0.019	0.137
Support Vector Regression	76.82%	41.19	6.41

Analysis

Linear Regression: With an R2 score of 99.99%, Linear Regression exhibits near-perfect accuracy, making it the most reliable model for predicting fantasy points. Its low MSE and RMSE scores further confirm its robustness and precision.

Decision Tree: Achieving an R2 score of 92.55%, the Decision Tree model also performs well, though slightly less accurate than Linear Regression. Its higher MSE and RMSE indicate a modest margin of error.

Support Vector Regression: While SVR shows reasonable predictive capability with an R2 score of 76.82%, it lags behind the other models. Its higher error metrics suggest that SVR may require more complex or fine-tuned data for optimal performance.

The evaluation phase underscores the efficacy of the models in predicting NBA player fantasy points. Linear Regression stands out as the top-performing model, followed by Decision Tree and Support Vector Regression. These results highlight the importance of choosing the appropriate algorithm based on the specific characteristics of the dataset and the goals of the analysis.

Through rigorous evaluation, this study confirms that these regression models can provide valuable insights and accurate predictions, essential for identifying potential MVP candidates and enhancing fantasy sports strategies.

F. Deployment

The deployment stage involves integrating the developed models into a web-based application. This step aims to make the research findings accessible to a broader audience and serve as a reference for future studies. The deployment is executed using Streamlit, allowing users to interact with the Decision Tree, Linear Regression, and Support Vector Regression models through the web application.

This deployment ensures that the research outputs are not only useful for current analysis but also provide a practical tool for ongoing and future basketball performance evaluations.

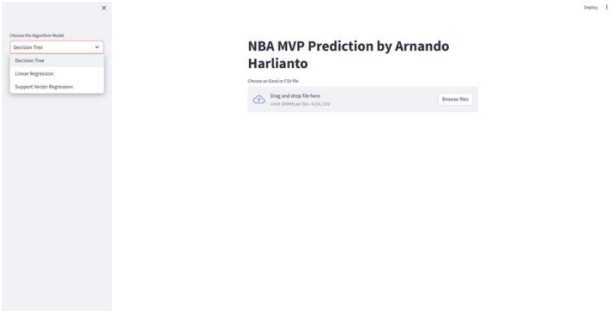


Fig. 2 NBA MVP Prediction

Fig. 2 shows the NBA Most Valueable Player Prediction Dashboard. This interface features a sidebar with options to select different algorithm models. Additionally, there is a file upload feature for Excel or CSV files. The uploaded files are processed by the selected algorithm model, producing results that include scatter plot visualizations, data tables of predictions, and R2 scores from the modeling process.



Fig. 3 – Visualization and prediction result

Fig. 3 displays the visualization and prediction results using the Decision Tree algorithm. To select Linear Regression or Support Vector Regression, use the options provided on the left side of Fig. 2.

V. Discussion

This study aimed to predict **fantasy points** for each NBA player using three regression algorithms: Decision Tree, Linear Regression, and Support Vector Regression (SVR). Each algorithm's performance was evaluated based on the R2 score and the prediction accuracy of fantasy points derived from various player statistics.

The models and their results are displayed through a web-based application built using Streamlit [17].

The analysis focused on comparing actual fantasy points earned by players (based on game performance metrics such as points, assists, rebounds, steals, and blocks) with the predicted fantasy points estimated by the machine learning models. By concentrating on the top 20 players with the highest fantasy points, the study ensured a stable dataset less affected by high variability. This approach facilitated clearer and more understandable presentations, allowing for a more accurate assessment of the prediction models against consistently high-performing players in fantasy sports.

Feature Importance Analysis

A detailed examination of feature importance across the three algorithms reveals valuable insights into the factors driving player performance. The Decision Tree algorithm identified "minutes played" and "field goals made" as the most critical features, emphasizing the significance of a player's presence on the court and their scoring efficiency. Other notable features included "free throw attempts," "double-doubles," "assists," "defensive rebounds," "points," and "total rebounds." These features collectively underscore the multifaceted contributions of players in various aspects of the game.

Support Vector Regression highlighted "points," "assists," and "rebounds" as primary features, with additional importance placed on "steals," "blocks," and "turnovers." This indicates that both offensive and defensive capabilities are crucial for achieving high fantasy points. The algorithm's emphasis on assists and rebounds aligns with the broader understanding that versatile players who contribute in multiple areas tend to be more valuable in fantasy sports.

Linear Regression placed the highest importance on "blocks," "steals," "assists," and "rebounds," followed by "turnovers," "points," "field goals made," "free throws made," and "three-point shots made." This distribution highlights the critical role of defensive actions and overall playmaking ability in determining a player's fantasy points. By considering a comprehensive range of performance metrics, Linear Regression provides a nuanced view of player contributions.

Comparative Performance

The comparative performance analysis of the algorithms is crucial for understanding their predictive power and practical applications as shown in Table 3. The Linear Regression model, with an R2 score of 99%, demonstrated near-perfect accuracy, making it the most reliable predictor of fantasy points. Its minimal Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) scores further validate its precision and robustness.

Decision Tree, with an R2 score of 92.55%, also performed well, although it did not match the high accuracy of Linear Regression. Its higher MSE and RMSE scores indicate a greater margin of error, suggesting that while it is a strong performer, it may not capture all the nuances of the data as effectively as Linear Regression.

Support Vector Regression, despite having the lowest R2 score of 76.82%, still provided valuable insights. Its performance highlights the importance of selecting appropriate algorithms based on the dataset's complexity and the specific features being analyzed. SVR's higher error metrics indicate it may require more complex data to achieve optimal performance, as suggested by previous studies.

These findings are consistent with studies [9] and [10], where Linear Regression outperformed Decision Tree. Conversely, studies [11] and [12] found that SVR performed better with more complex datasets, emphasizing the need for appropriate algorithm selection based on dataset properties. Despite previous research where Decision Tree surpassed Linear Regression, this study found Linear Regression to be nearly perfect in performance, underscoring the unique influence of dataset characteristics on each algorithm's effectiveness.

VI. Conclusion

This study effectively employed three regression algorithms—Linear Regression, Decision Tree, and Support Vector Regression (SVR)—to predict fantasy points for NBA players based on the 2022/2023 season data. The analysis demonstrated that fantasy points, derived from cumulative statistics such as points, rebounds, assists, steals, and blocks, serve as a reliable benchmark for identifying potential MVP candidates. These findings reinforce the importance of algorithm selection based on dataset properties to achieve optimal predictive performance in basketball analytics.

Limitations of the Study

Several limitations were encountered during this study:

Dataset Scope: The dataset was limited to regular-season games, excluding playoff and preseason matches. This exclusion aimed to maintain data consistency but may overlook performance variations in different game contexts.

Data Curation: Curating the dataset to include only active players for the 2022/2023 season required meticulous filtering, which might have inadvertently excluded relevant data points that could have influenced the models' accuracy.

Algorithm Constraints: The study's focus was on three specific regression algorithms. Other potentially effective algorithms were not explored, which may have provided different insights and enhanced predictive performance.

Subjective Factors: The study primarily relied on quantitative data, while subjective factors such as media voting and fan influence, which play a crucial role in MVP selection, were not incorporated.

Contributions

This research contributes significantly to the field of sports analytics, particularly in the following areas:

Algorithmic Insights: It provides a comparative analysis of three regression algorithms, highlighting Linear Regression's superior performance in predicting NBA player fantasy points.

Feature Importance: The study offers a detailed examination of feature importance across different algorithms, shedding light on key performance metrics that drive player success.

Practical Application: By deploying the predictive models in a web-based application using Streamlit, the research enhances accessibility and usability for analysts and enthusiasts, facilitating informed decision-making in fantasy sports and MVP predictions.

Future Research

Future research directions include:

Incorporating Qualitative Factors: Integrating qualitative data such as media voting, fan influence, and player off-court behavior could provide a more holistic approach to MVP prediction.

Exploring Additional Algorithms: Investigating other machine learning algorithms, including ensemble methods and deep learning techniques, could further improve predictive accuracy and robustness.

Expanding the Dataset: Including multiple seasons and playoff performances could offer a more comprehensive view of player capabilities and enhance the models' generalizability.

Advanced Statistical Techniques: Applying advanced statistical techniques and feature engineering could uncover deeper insights and refine the predictive models.

Real-Time Predictions: Developing models capable of real-time predictions and updates based on live game data could revolutionize fantasy sports and MVP forecasting.

By addressing these areas, future research can build on the findings of this study, advancing the capabilities of sports analytics and enhancing our understanding of player performance and MVP selection in professional basketball.

ACKNOWLEDGMENT

We gratefully acknowledge the support from Big Data Lab Univesitas Multimedia Nusantara for this research.

REFERENCES

[1] J. Han and Z. Yu, "Random Forest Prediction of NBA Regular Season MVP Winners Based on Metrics Optimization," *Inf. Knowl. Manag.*, vol. 4, no. 4, pp. 53–62, 2023, doi: 10.23977/infkm.2023.040409.

[2] M. Manisera, R. Metulini, and P. Zuccolotto, "Basketball analytics using spatial tracking data," in *Springer Proceedings in Mathematics and Statistics*, 2019, vol. 288, doi: 10.1007/978-3-030-21158-5_23.

[3] J. V. R. da Silva and P. C. Rodrigues, "All-NBA Teams' Selection Based on Unsupervised Learning," *Stats*, vol. 5, no. 1, 2022, doi: 10.3390/stats5010011.

[4] A. A. Albert, L. F. de Mingo López, K. Allbright, and N. Gómez Blas, "A Hybrid Machine Learning Model for Predicting USA NBA All-Stars," *Electronics*, vol. 11, no. 1, p. 97, Dec. 2021, doi: 10.3390/electronics11010097.

[5] B. Mahesh, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, vol. 9, no. 1, 2020.

[6] S. Badillo *et al.*, "An Introduction to Machine Learning," *Clin. Pharmacol. Ther.*, vol. 107, no. 4, 2020, doi: 10.1002/cpt.1796.

[7] R. Sharma, "Study of Supervised Learning and Unsupervised Learning," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 6, 2020, doi: 10.22214/ijraset.2020.6095.

[8] A. Possolo, A. Koepke, D. Newton, and M. R. Winchester, "Decision tree for key comparisons," *J. Res. Natl. Inst. Stand. Technol.*, vol. 126, 2021, doi: 10.6028/jres.126.007.

[9] A. Karim, "Perbandingan Prediksi Kemiskinan di Indonesia Menggunakan Support Vector Machine (SVM) dengan Regresi Linear," *J. Sains Mat. dan Stat.*, vol. 6, no. 1, 2020, doi: 10.24014/jsms.v6i1.9259.

[10] I. Oktavianti, E. Ermatita, and D. P. Rini, "Analisis Pola Prediksi Data Time Series menggunakan Support Vector Regression, Multilayer Perceptron, dan Regresi Linear Sederhana," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, 2019, doi: 10.29207/resti.v3i2.1013.

[11] L. M. Ginting, M. M. Sigiro, E. D. Manurung, and J. J. P. Sinurat, "Perbandingan Metode Algoritma Support Vector Regression dan Multiple Linear Regression Untuk Memprediksi Stok Obat," *J. Appl. Technol. Informatics Indones.*, vol. 1, no. 2, 2021, doi: 10.54074/jati.v1i2.36.

[12] J. Setiawan, A. Milenia, and A. Faza, "An Integrated Approach for Sentiment Analysis and Topic Modeling of a Digital Bank in Indonesia using Naïve Bayes and Latent Dirichlet Allocation Algorithms on Social Media Data," in *2023 4th International Conference on Big Data Analytics and Practices, IBDAP 2023*, 2023, doi: 10.1109/IBDAP58581.2023.10271956.

[13] R. Karim, M. K. Alam, and M. R. Hossain, "Stock market analysis using linear regression and decision tree regression," in *2021 1st International Conference on Emerging Smart Technologies and Applications, eSmarTA 2021*, 2021, doi: 10.1109/eSmarTA52612.2021.9515762.

[14] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," in *Procedia Computer Science*, 2021, vol. 181, doi: 10.1016/j.procs.2021.01.199.

[15] F. Anwar and J. Setiawan, "Unleashing the Power of Time-Series Method: Iluminating Indonesia's Open Unemployment Rate Based on Educational Attainment," in *Proceedings of the 7th 2023 International Conference on New Media Studies, CONMEDIA 2023*, 2023, doi: 10.1109/CONMEDIA60526.2023.10428661.

[16] ESPN, "Fantasy basketball - Who is the real fantasy MVP this season? - ESPN." [Online]. Available: https://www.espn.com/fantasy/basketball/story/_/id/26046899/fantasy-basketball-real-fantasy-mvp-season. [Accessed: 13-Jun-2024].

[17] "Streamlit • A faster way to build and share data apps." [Online]. Available: https://streamlit.io/. [Accessed: 14-Jun-2024].

turnitin Page 108 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Adverse Media Classification: A New Era of Risk Management with XGBoost and Gradient Boosting Algorithms

1st Reza Juliandri
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
reza.juliandri@student.umn.ac.id

2nd Monika Evelin Johan
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

3rd Jansen Wiratama
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
jansen.wiratama@umn.ac.id

4th Samuel Ady Sanjaya
Information System
Universitas Multimedia Nusantara
Tangerang, Indonesia
samuel.ady@umn.ac.id

***Abstract*— Adverse media is negative information that is not profitable for businesses or individuals, while adverse media classification is the process of classifying news titles that are included in adverse media. In an effort to create a system capable of mitigating the occurrence of fraud for customer satisfaction, machine learning is used to classify news both as detrimental media and not for the selection of news for the customer due diligence system. This study utilizes the XGBoost and Gradient Boosting algorithms to classify news headlines. A data set of 1,281 records was collected from NewsAPI and web scraping. Back translation is used in the data preparation stage to deal with unbalanced data sets and create text variants Grid search is used to find the best hyperparameters for Gradient Boosting and XGBoost. The results of the research are in the form of a machine-learning model. Across all models examined, Gradient Boosting trained on 753 records performed best with an accuracy rate of 82.31% on test data and 84.93% on validation data. This model is able to be used to classify media and then implemented in a web-based interface.**

***Keywords*— adverse media, classification, gradient boosting, website, XGBoost.**

I. Introduction

The proliferation of Financial Technology (Fintech) in Indonesia has been notably swift, coinciding with the society's growing acceptance of digital payments. In 2021, there was a 32.5% augmentation in the number of financial technology enterprises relative to the preceding year [1]. The data from 2022 indicates a rise in electronic money transactions, registering a growth of 42.06% in the first quarter alone [2]. Furthermore, the financial technology sector in Indonesia manifests a Compounded Annual Growth Rate (CAGR) of 39% [3].

Technology has given rise to financial technology (fintech) innovations that make it possible to carry out transaction activities quickly and efficiently by utilizing technology [4]. The rapid development of financial technology also increases the risk of money laundering and terrorism financing, especially in this industry. The fast and dynamic nature of financial technology makes financial technology a medium for money laundering and terrorism financing [5]. This causes financial technology companies to be required to identify, verify and analyze user risks in implementing the Anti-Money Laundering and Counter-Terrorism Financing (AML-CTF) program[6].

A start-up company can try to develop similar technology oriented towards solving fraud problems and customer trust problems [7]. In improving customer due diligence services, a machine learning model is needed to classify data from media, especially from online media in Indonesia. The adverse media model created will be used to sort data before it is processed in the customer due diligence system. The data used in making this model comes from scraping on news sites and using NewsAPI as the data source for the model to be made.

The research utilizes XGBoost, known for its ability to handle large data quickly and manage missing values and data imbalance efficiently. However, XGBoost may require more time to train the model and a deeper understanding of its parameters to achieve optimal results. The study will also use Gradient Boosting for adverse media classification to determine if XGBoost outperforms its predecessor.

The ultimate goal is to create a model for classifying news into adverse or non-adverse media categories, assisting a company in processing customer due diligence data sources with higher accuracy. The study aims to address the limitations of previous research, particularly in adverse media classification in Indonesia.

II. Related Works

Several previous studies have focused on text classification related to news data using the AG News dataset. The first algorithm for news text classification is XLNet, achieving an error rate of 4.45% [8]. This research, conducted by Carnegie Mellon University and the Google AI Brain Team, reported an accuracy rate of 85.4%. Another study in text classification used BERT-ITPT-FiT, with an error rate of 4.8% [9]. A subsequent study employed LSTM with a Mixed Objective Function, resulting in an error rate of 4.95% [10]. Another study utilized ULMFiT, achieving an error rate of 5.01% [11].

There are prior studies on text classification using machine learning. One such study employed the Random Forest algorithm with Gradient Boosting for automated complaint categorization [12]. This research successfully applied Random Forest and Gradient Boosting for multi-class

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

text classification, with an accuracy of 73%. The limitation of this study was that most data were classified into a single category due to the words falling into multiple categories.

Another prior study used Decision Tree and XGBoost for COVID-19 vaccine sentiment classification [13]. This research produced an XGBoost model with an accuracy rate of 66%, and a Decision Tree model with an accuracy of 65%. The XGBoost algorithm outperformed the Decision Tree algorithm on an imbalanced dataset. This study had the limitation of an imbalanced dataset and achieved a low accuracy and f1-score of 66%.

The next study was on fake news classification using the XGBoost algorithm [14]. This research achieved an accuracy rate of 92% with an 80:20 dataset ratio. This level of accuracy was obtained based on parameter tuning. The limitation of this study was that the dataset contained only 100 randomly distributed data points. The research may only have the best accuracy on specific datasets and may not produce the same accuracy level on different datasets.

Another study predicted chronic kidney failure by comparing Grid Search and Random Search methods for hyperparameter tuning in the XGBoost algorithm[15]. This study achieved an accuracy rate of 99.29% and an f-measure of 0.99. XGBoost with hyperparameter tuning, z-score normalization, and random oversampling played a significant role in achieving this accuracy. The drawback of this study was the lengthy time taken to find the correct hyperparameters using Grid Search. In addition, the study was not tested on different types of datasets, so the generalization ability of the model was unknown.

Another study predicted and diagnosed future diabetes risk using machine learning. This study produced a model with an accuracy rate of 85% using Gradient Boosting, 77% with Naïve Bayes, and 79% with Logistic Regression[16]. This research achieved high predictive accuracy and used algorithms that are excellent for regression problems, such as Gradient Boosting. This research had several limitations, such as the fact that the data used did not cover the entire population. The research only used BMI and plasma glucose, which are crucial in predicting diabetes. The study has potential overfitting issues because it does not address potential overfitting that can occur in the model.

Based on the results of previous research using machine learning algorithms like Random Forest, convolutional neural network, complement, multinomial naïve bayes, Linear Regression, and XGBoost, there is currently no research on creating models for adverse media, particularly in the Indonesian language. Most of the research on customer due diligence and adverse media is conducted privately. Moreover, this research uses datasets that are not publicly available and requires scraping techniques and API calls for data collection. Therefore, this study will focus on creating a machine learning model used for classifying news, especially online media, whether the content of the news falls into adverse media or not. This study also tests the use of TF-IDF as in research [18], as well as the use of CountVectorizer and Tokenizer, which are commonly used in text data feature extraction. The use of Grid Search in this study aims to determine the best parameters more efficiently. Grid Search facilitates the discovery of optimal hyperparameters in the algorithms used. The parameters will influence the level of accuracy as revealed in studies [14], [15].

III. METHODOLOGY

This research will create a machine learning model to classify news and identify whether the news is classified as adverse media or not. The algorithms used in model training are XGBoost and Gradient Boosting. The research methodology is depicted in the flowchart presented in Fig. 1.

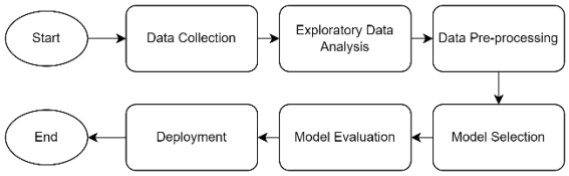


Fig 1. Research Flow

A. Data Collection

Data is gathered using two methods: web scraping and the NewsAPI. The collected news data will be sorted based on keywords that indicate adverse media. The keywords used are based on guidelines from the Indonesian Center for Reporting and Analysis of Financial Transactions (Pusat Pelaporan dan Analisis Transaksi Keuangan (PPATK)). The total data generated is 1,281 records originating from the keywords "korupsi" (corruption), "suap" (bribe) and "narkotika" (narcotics). The data collection process was from 24 January 2022 to 6 February 2022.

B. Exploratory Data Analysis

Exploratory data analysis in this study is by using the univariate analysis method. The analysis is carried out using news title data by calculating the frequency of words from the available dataset. This analysis also looks at the words that appear the most in the news headlines and the words that appear the least often in the news headlines. Apart from frequency, another analysis is by making histograms and word clouds to find out more about the data used.

C. Data Pre-processing

This stage is the stage in the process of cleaning the data and turning the news text into a token matrix. In the process of cleaning the data, the data obtained must be cleaned of some characters that are not needed. After that, a case folding process is carried out which has the goal of turning all letters into lowercase letters [17]. Besides that, the process of cleaning the data also needs to remove stop words so that they don't interfere with the training process in model making. Next stage is to perform the feature extraction process using several types, namely TF-IDF, Tokenizer, and CountVectorizer.

D. Model Selection

According to the previous research, it stated that there is no model that is generally suitable for certain data and purposes. The process of finding a model that fits the dataset that is owned is done by finding the most suitable model or in accordance with the characteristics of the existing data [19]. Model selection is used to get the best model according to the data provided. The model selection process in XGBoost and Gradient Boosting is done by looking for the best hyperparameter with the best performance. The process of finding the best hyperparameters in XGBoost and Gradient Boosting can be done using a grid search. XGBoost has several hyperparameters that can be used. Some of these

parameters include learning_rate, max_depth, min_child_weight and gamma [18].

E. Model Evaluation

Model evaluation is done by comparing the average precision and recall values between the train data and test data or validation data. If the difference exceeds 15%, the model is overfitting. If the difference is less than 1%, the model is underfitting. The optimal model has a difference between 1% and 5%. The calculation process from the difference between precision and recall and generates Good Fit, Overfit and Underfit labels. The resulting model is as shown in table 1.

TABLE I. MODEL RESULTS

Dataset	Algorithm	Accuracy	Fit 5%	Fit 15%
753	Gradient Boosting - CountVectorizer	82.31%	Overfit	Good Fit
1.281	XGBoost – TF-IDF	78.54%	Overfit	Good Fit

The top-performing model derived from XGBoost demonstrates a robust performance, incorporates TF-IDF text processing, and makes use of a dataset with 1,281 instances. With regard to test data and validation data, this model achieves an admirable accuracy of 78.54% and 82.52%, respectively. It was chosen because it strikes a better balance between model stability and performance. In particular, it performs better than alternatives in terms of metrics like precision, recall, F1-score, and overall accuracy. Due to overfitting concerns in models using CountVectorizer under various dataset conditions, a problem that was particularly noticeable within the XGBoost framework, the use of TF-IDF emerged as a viable option.

Moving on to the area of Gradient Boosting, the top model uses CountVectorizer for text processing and runs on a dataset with 753 instances. This model stands out because it performs at its best across a variety of evaluative criteria, such as accuracy, precision, recall, and F1-score. Despite exhibiting slight overfitting symptoms—encountered at the 5% limit but avoided at the 15%—the model maintains its usefulness in practice due to its reliable and outstanding performance. It produces test data accuracy rates of 82.31% and validation data accuracy rates of 84.93%.

The Gradient Boosting model, distinguished by its accuracy rate of 82.31% on test data and 84.93% on validation data, is chosen for the deployment phase after a thorough comparison of the two models. The deployed model will be used to categorize news titles into negative media categories, demonstrating a robust use case for the model. The selected model is ready for integration with the Flask framework at the deployment stage.

F. Deployment

Deployment is carried out using a website platform connected to the Adverse Media Classification API. The backend of Adverse Media Classification is built using the Python Flask framework. The Flask framework handles request from the frontend, routing them to the designed endpoints for processing. The backend subsequently validates data and executes the specified algorithm. The backend invokes the model and the responses from the model then converted into JSON values to utilization on the frontend page. The backend requires several parameters in using the Adverse Media Classification model. The parameters used are title and algorithm. The title parameter will be used as the title of the news that will be analyzed with the model that has been made. Algorithm parameters are used to select the algorithm used in the classification process. The website platform can be accessed via the frontend once it has been deployed. Users can access the website from anywhere, provided they have an internet connection and the platform is deployed on the server.

IV. RESULTS AND DISCUSSION

This study offers a comparative evaluation of machine learning models for media classification with a specific focus on adverse and non-adverse categories. Gradient Boosting and XGBoost models were investigated thoroughly, with the former exhibiting superiority in terms of accuracy, precision, recall, f1-score, and train-test differences. The achieved accuracy rate for most models exceeded 70%.

The study employed diverse text processing techniques such as TF-IDF, CountVectorizer, and Tokenizer, yielding varying results. Despite the Tokenizer's high accuracy in certain models, its performance was subpar on test data due to overfitting, suggesting the necessity for further investigation. Interestingly, the study discovered that there is no direct correlation between data volume and model accuracy. This suggests that increasing the amount of data does not necessarily result in decreased accuracy. The research also underscored the idea that the accuracy achieved in machine learning models may not always represent real-world accuracy due to limitations such as data availability and variations in word usage.

These findings align with previous studies. For example, the Gradient Boosting model outperformed the Random Forest in a study titled "Automatic Complaints Categorization Using Random Forest and Gradient Boosting"[12], a result that was replicated in the current study. Likewise, the effectiveness of Grid Search, which was used in the current study, was confirmed in "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure"[15]. The practical application of this research is the creation of a website for news title analysis. The study recommends continuous monitoring of model performance to guarantee its applicability. Furthermore, there is potential for future enhancements and the integration of the model into company's customer due diligence systems via an API.

V. CONCLUSION

The accuracy of the produced model in classifying adverse media and non-adverse media becomes a measure of the effectiveness of the generated model. The classification model was created using the Python programming language, applying the XGBoost and Gradient Boosting algorithms. The result of the created model will be embedded in a Flask framework, and users can utilize this model through a web page or API call. The model used is a Gradient Boosting model with an accuracy rate of 82.31% on test data and 84.93% on validation data.

The model from this research can be used to classify adverse media on new, which this classification process will simplify the task of grouping the news that will be used in the customer due diligence system. This model is already usable

in the classification process but still needs further development due to the limitations of the data used.

The generated model is a model from the Gradient Boosting algorithm with an accuracy value of 82.31% on test data and 84.93% from validation data. Although these models have a sufficient accuracy rate, it should be remembered that the limitations of the data used in this study affect the model's performance. The amount of data used is only as much as 753, which is still considered insufficient. Therefore, to improve the performance of this model, more data and more complex word combination variations are required.

In addition, these models may not effectively deal with new data that has never been taught during training. The variation of circulating news titles is very diverse, and this model may not be able to handle all these variations with a high accuracy rate. Therefore, this research recognizes these limitations and suggests using more data in future research to improve model performance.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Universitas Multimedia Nusantara for their invaluable support, which played a pivotal role in the successful completion of this research endeavor. Their substantial contribution was instrumental in achieving our objectives, and we are deeply grateful for their unwavering assistance.

REFERENCES

[1] A. Karnadi, "Jumlah Fintech di Indonesia Terus Meningkat hingga 2021." Accessed: Mar. 15, 2023. [Online]. Available: https://dataindonesia.id/digital/detail/jumlah-fintech-di-indonesia-terus-meningkat-hingga-2021

[2] D. F. Rahman, "Transaksi Keuangan Digital Tumbuh Pesat pada Triwulan I 2022." Accessed: Mar. 15, 2023. [Online]. Available: https://databoks.katadata.co.id/datapublish/2022/04/19/transaksi-keuangan-digital-tumbuh-pesat-pada-triwulan-i-2022

[3] "Bulan Fintech Nasional, 1,5 Juta Masyarakat Berpartisipasi dan Dapatkan Edukasi Fintech dari Pemerintah, Asosiasi, dan Pelaku Industri." Accessed: Mar. 15, 2023. [Online]. Available: https://www.bi.go.id/id/publikasi/ruang-media/news-release/Pages/sp_2433922.aspx

[4] E. N. Sugiarti, N. Diana, and M. C. Mawardi, "PERAN FINTECH DALAM MENINGKATKAN LITERASI KEUANGAN PADA USAHA MIKRO KECIL MENENGAH DI MALANG," e_Jurnal Ilmiah Riset Akuntansi, vol. 8, no. 04, Jul. 2019, Accessed: Jan. 07, 2023. [Online]. Available: http://riset.unisma.ac.id/index.php/jra/article/view/4038

[5] H. M. A. Nasution, "HATI-HATI PENCUCIAN UANG DI INDUSTRI FINTECH! - PUSAT PELAPORAN DAN ANALISIS TRANSAKSI KEUANGAN." Accessed: Mar. 15, 2023. [Online]. Available: https://www.ppatk.go.id/siaran_pers/read/969/hati-hati-pencucian-uang-diindustri-fintech.html

[6] F. Annisa and P. R. Putri, "PENERAPAN PROGRAM APU PPT UNTUK MENCEGAH PENCUCIAN UANG DAN PENDANAAN TERORISME PADA INDUSTRI FINTECH," ADIL: Jurnal Hukum, vol. 11, no. 2, Dec. 2020, doi: 10.33476/AJL.

[7] "Tentang Kami — Kredibel." Accessed: Jan. 07, 2023. [Online]. Available: https://www.kredibel.co.id/about

[8] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," Jun. 2019, [Online]. Available: http://arxiv.org/abs/1906.08237

[9] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to Fine-Tune BERT for Text Classification?," May 2019, [Online]. Available: http://arxiv.org/abs/1905.05583

[10] D. S. Sachan, M. Zaheer, and R. Salakhutdinov, "Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function," Sep. 2020, doi: 10.1609/aaai.v33i01.33016940.

[11] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," Jan. 2018, [Online]. Available: http://arxiv.org/abs/1801.06146

[12] M. T. Anwar, "Automatic Complaints Categorization Using Random Forest and Gradient Boosting," Advance Sustainable Science, Engineering and Technology, vol. 3, no. 1, p. 0210106, Apr. 2021, doi: 10.26877/asset.v3i1.8460.

[13] H. H. Sinaga and S. Agustian, "Pebandingan Metode Decision Tree dan XGBoost untuk Klasifikasi Sentimen Vaksin Covid-19 di Twitter," Jurnal Nasional Teknologi dan Sistem Informasi, vol. 8, no. 3, pp. 107–114, Dec. 2022, doi: 10.25077/TEKNOSI.v8i3.2022.107-114.

[14] J. P. Haumahu, S. D. H. Permana, and Y. Yaddarabullah, "Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost)," IOP Conf Ser Mater Sci Eng, vol. 1098, no. 5, p. 052081, Mar. 2021, doi: 10.1088/1757-899x/1098/5/052081.

[15] D. A. Anggoro and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," International Journal of Intelligent Engineering and Systems, vol. 14, no. 6, pp. 198–207, Dec. 2021, doi: 10.22266/ijies2021.1231.19.

[16] R. Birjais, A. K. Mourya, R. Chauhan, and H. Kaur, "Prediction and diagnosis of future diabetes risk: a machine learning approach," SN Appl Sci, vol. 1, no. 9, Sep. 2019, doi: 10.1007/s42452-019-1117-9.

[17] M. N. Randhika, J. C. Young, A. Suryadibrata, and H. Mandala, "Implementasi Algoritma Complement dan Multinomial Naïve Bayes Classifier Pada Klasifikasi Kategori Berita Media Online," Ultimatics : Jurnal Teknik Informatika, vol. 13, no. 1, pp. 19–25, Jun. 2021, doi: 10.31937/TI.V13I1.1921.

[18] Y. Wang and X. S. Ni, "A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization," International Journal of Database Management Systems, vol. 11, no. 01, pp. 01–17, Jan. 2019, doi: 10.48550/arxiv.1901.08433.

[19] Frentzen, J. Wiratama and R. S. Oetama, "KNN And Naïve Bayes Algorithms for Improving Prediction of Indonesian Film Ratings using Feature Selection Techniques," 2023 4th International Conference on Big Data Analytics and Practices (IBDAP), Bangkok, Thailand, 2023, pp. 1-6, doi: 10.1109/IBDAP58581.2023.10271977.

turnitin Page 112 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Evaluating GRNN, Decision Tree, and Random Forest: A Gas Turbine Emission Prediction Comparative Study

Rudy Winarto
School of Interdisciplinary Management and Technology
Sepuluh Nopember Institute of Technology
Surabaya, Indonesia
6047222026@mhs.its.ac.id

Mauridhi Hery Purnomo
Department of Electrical Engineering
Sepuluh Nopember Institute of Technology
Surabaya, Indonesia
hery@ee.its.ac.id

Wiwik Anggraeni
Department of Information System
Sepuluh Nopember Institute of Technology
Surabaya, Indonesia
wiwik@is.its.ac.id

***Abstract*— The exhaust CO, CO_2, O_2, SO_2, and NO gas emissions of a gas turbine-powered compressor unit under specific operating conditions are studied. Due to the high costs of hardware, maintenance, and calibration, Predictive Emissions Monitoring Systems (PEMS) is a more potential alternative to traditional CEMS for monitoring gas turbine emissions. PEMS provides a cost-effective and precise solution to traditional hardware-based emissions monitoring by employing algorithms to predict emissions. Market PEMS models use empirical approach modeling with limited data. This research explores Decision Trees and Random Forest for a new model that can handle more data of multiple inputs and output and compares its performance to the GRNN modelling approach. This study analyzed one million data points on gas turbine emissions (collected from 2021 to 2023) and found Random Forest to be the most accurate prediction method, while Decision Tree offers a good balance for smaller datasets, and Generalized Regression Neural Network (GRNN) is best for simpler data.**

Keywords—Decision Tree, Emission, GRNN, Prediction, Random Forest

I. INTRODUCTION

Gas turbines are essential for the process industry, power generations, and transportation. However, it also contributes to emissions by releasing pollutants like Nitrogen Oxide (NO), Carbon Monoxide (CO), Sulphur Dioxide (SO_2), and other particulate matter [1],[2].

Continuous Emissions Monitoring Systems (CEMS) are widely used as monitoring devices for gas turbine emissions. To sustain its level of accuracy, CEMS requires substantial costs for acquiring hardware, ongoing maintenance, and regular calibration of sensors. Predictive Emissions Monitoring Systems (PEMS) have emerged as viable alternatives to Continuous Emissions Monitoring Systems (CEMS) for various process industries to mitigate expenses related to emissions monitoring. This system does not require installing equipment such as CEMS and can use the available field process data.

Based on the available literature, it has been observed that the development of PEMS in the market has predominantly utilized a non-linear regression mathematical model approach and a simple artificial neural network method. However, the predicted results obtained through these methodologies have been discovered to show a significant deviation from the original data pattern.

II. EMISSIONS PREDICTION OVERVIEW

A. Gas Turbine

The gas turbine is a type of internal combustion engine that consists of several critical elements, such as a compressor, combustion chamber, turbine, and even a power turbine. Incoming air is compressed by the compressor before being mixed with fuel in the combustion chamber, where it ignites. The main turbine is powered by high-velocity exhaust gases, which generate mechanical energy for a variety of purposes, including aircraft propulsion and electricity generation.

B. Emissions Predictions Researches

A specific PEMS model of a natural gas turbine-driven compressor was developed by [3] in 2011 using optimized Neural Network architecture to predict the emissions of NOx (multiple inputs, single outputs), a grey box modelling. The neural network is based on feed-forward architectures with back-propagation optimization. As a result, the input variables uncertainty contributed to 0.9-3% of NOx (ppmv) and 3.5-6% of NOx (kg/hr).

In 2019, [3], [4] is predicting the emissions of gas turbines using machine learning for classification and regression called ELMs (Extreme Learning Machines). The total dataset used is 36,733 over 5 (five) years of operations. It has been observed that predictive performance is related to the strong level of correlation (CO compared to NOx predictive case). Dataset size will correlate to the maximum mean absolute error.

A model to predict the NOx emissions on the natural gas-fired cogeneration unit has been developed by [5] involving 12,086 examples of datasets and 3,020 testing sets. They used a neural network-based model with 8 (eight) input variables to predict a single NOx as output. Using the test dataset, the model resulted in a 0.14% difference between measured and predicted values.

III. METHODOLOGY

According to [6], process data collection, model development, validation, and testing are the main activities of PEMS implementation. To ensure the highest reliability and robustness of the final models, the data collection phase must be precisely designed and executed to involve all typical operating scenarios [7].

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

turnitin Page 113 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

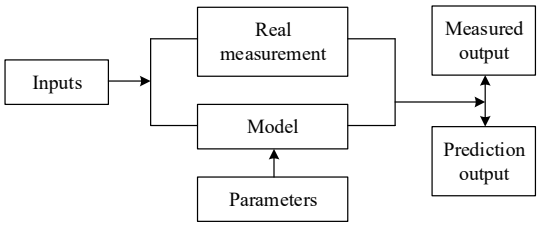


Fig. 1. Emission prediction development approach

Model creation and test are essential parts of any emission prediction development. This involves multiple tasks, from raw data preparation to the model's final testing. These tasks require applying advanced statistical, mathematical, and modelling techniques.

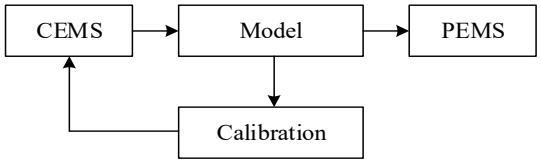


Fig. 2. Model calibration scheme

The Fig. 3 shows a PEMS (Predictive Emissions Monitoring System) model. Inputs like gas flow, fuel, and various process and turbine engine's parameters readings are fed into the model. The model then uses these inputs to predict emission levels of CO, CO_2, O_2, SO_2, and NO. The diagram illustrates how PEMS uses operational data to predict emissions.

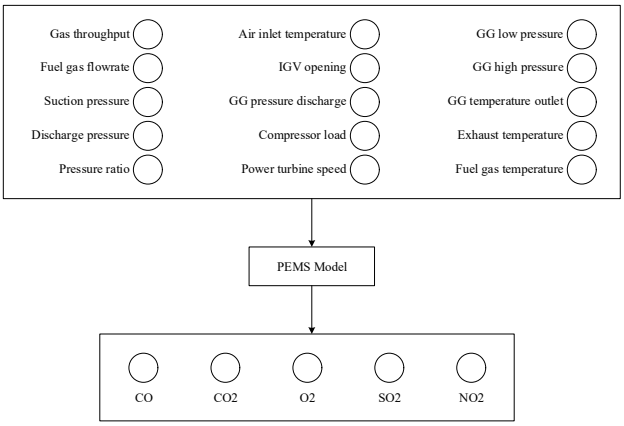


Fig. 3. PEMS model with input and output variables

A. Generalized Regression Neural Network (GRNN)

The Generalized Regression Neural Network (GRNN), the typical schematic shown in Fig. 4, is a modified version of radial basis neural networks. The suggestion of using GRNN was made by D.F. Specht in 1991[8].

GRNN is an adaptable tool that can be employed for regression, prediction, and classification tasks. The Generalized Regression Neural Network (GRNN) is an appropriate choice for effectively addressing online dynamical systems [9]. GRNN is an enhanced method in neural networks that is based on nonparametric regression. The concept is that each training sample will serve as a reference point for a radial basis neuron [10].

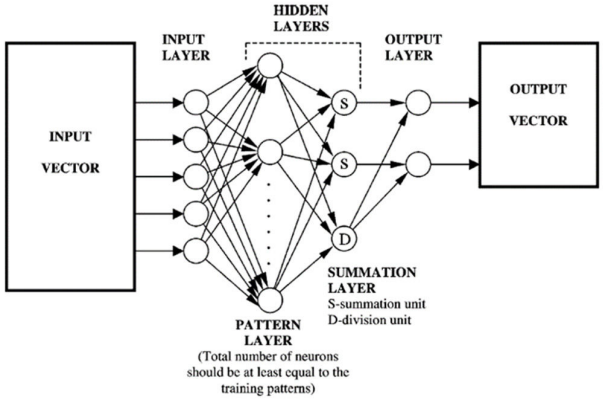


Fig. 4. Typical schematic diagram of GRNN[10]

Python script is used to perform GRNN modelling using the sklearn library. It begins with importation of the required libraries and metrics which will be used to calculate the model performance. A regression neural network model is generated and then train this model using the standard scaled training data. The model makes predictions on the scaled test data after this training process. The Phyton script is built around three basic routines: linear regression, model training, prediction of new data with the model.

Using a setup with 16 input parameters, a 32-neuron hidden layer with ReLU activation, a single output neuron with linear activation, and 100 epochs of Adam's optimization, the GRNN method is applied as a predictive model.

B. Decission Tree

A decision tree is a crucial tool in machine learning, functioning as a supervised learning method [11]. A decision tree is so named because it resembles an actual tree. Beginning with a single root node, it branches out to internal nodes in response to decisions made by the system (questionnaires regarding the features of the data). The branches ultimately extend to leaf nodes, symbolizing the ultimate forecasts or classifications. Due to the flowchart-like structure that simplifies comprehension of the reasoning behind the model's decisions, decision trees are widely used.

This is a significant advantage over other machine learning algorithms; it can be applied to both classification and regression tasks (predicting discrete categories and continuous values, respectively). It is a well-established method for which numerous tools and algorithms are available [12], [13].

By using Python script, the implementation of a decision tree regression model is utilizing the sklearn library. It begins by creating an instance of Decision Tree Regressor. The model is trained on the scaled training data and predictions on the scaled test data are made.

C. Random Forest

An algorithm for machine learning, the random forest is an extension of the decision tree concept [14], [15], [16]. It is a supervised learning method, like decision trees, in which predictions are generated for unobserved data using information learned from labeled data. Random forests are an effective method due to their utilization of ensemble learning. In other words, they generate a single, more precise prediction by combining the outputs of numerous decision trees [17].

turnitin Page 114 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

The Python script will utilizes a Random Forest regressor from the sklearn library to make predictions. The process begins by importing the Random Forest Regressor and creating an instance of the model with number of estimators set to 10 and a random state of 30 to ensure reproducibility. The model is trained on the scaled training data. After training, the model predicts the target variable on the scaled test data.

D. Model Performance Measurement

According to [18], to assess the model's performance, mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and the R^2 score are calculated. These metrics provide a comprehensive evaluation of the model's accuracy and predictive power.

$$MSE = \frac{\sum_i^N (P_i - O_i)^2}{N} \quad (1)$$

$$MAE = \frac{\sum_i^N |P_i - O_i|}{N} \quad (2)$$

$$MAPE = \frac{1}{N}\left(\frac{\sum_i^N |P_i - O_i|}{O_i}\right) \quad (3)$$

E. Process Equipment Description

For the study, a compressor unit powered by a gas turbine is chosen. The suction pressure is 30 psig with a discharge pressure of 210 psig. The gas flow rate is designed to be 100 MMscfd and driven by a capacity of 39,520 horsepower (hp).

The compression system is equipped with a Continuous Emissions Monitoring System installed on the exhaust stack. This equipment is utilized to analyze the flue gas emissions profile, specifically Carbon Monoxide (CO), Carbon Dioxide (CO_2), Oxygen (O_2), Sulphur Dioxide (SO_2), and Nitrogen Oxide (NO).

F. Dataset Preparation

Turbine and CEMS emissions-related process data collected at 5-second intervals from 00:00 on January 1, 2021, to 23:59 on December 31, 2023, from Process Historian Data. The predictive model development should be based on consistent and dependable process variables. The procedure for choosing the process variables is as follows:

- To ensure the model's accuracy, a single tag has been chosen to eliminate the exact process variable duplication [5].
- Identified potential process variables that contributed to emission pollutants productions, did a correlation analysis, and removed process variables unrelated to the production of emissions pollutants.
- Eliminated process variables with nearly zero variance do not significantly improve prediction.
- Collected data of continuous emissions monitoring system or process equipment downtime in specified maintenance periods were removed. This includes data rows with missing values that were removed.
- Discharged or eliminated process data corresponding to the possibility of instrumentation noise measurements, such as negative value, over-range, and under-range spike value, etc.).

Sixteen process variables were preliminary selected post the completion of the above steps as the input of model development, which are gas throughput, fuel gas flowrate, suction pressure, discharge pressure, pressure ratio, combustion air inlet temperature, IGV opening, GG pressure discharge, compressor load, power turbine speed, GG low pressure, GG high pressure, GG temperature outlet, exhaust temperature, and fuel gas temperature. An individual total of 50,337 data of each process variable, including emissions pollutants variables (CO, CO_2, O_2, SO_2, and NO).

A total of 1,006,740 data sets will be employed for emission prediction model development.

TABLE I. DATASET ALLOCATION

Dataset	Data Numbers
Training (80%)	805,392
Test (20%)	201,348
Total – Data	1,006,740

The data set was partitioned into a training set (80%) and a test set (20%), following a random shuffling process. The training set comprised the data utilized to construct models and optimize weights and biases for specific network structures.

G. Preliminary Data Analysis

The emission basic statistical data is revealed in TABLE II. It presented the dataset numbers of each variable, mean, minimum, and maximum range.

TABLE II. BASIC STATISTICAL DATA INFORMATION

Variable	Abbreviation	Unit	Min	Mean	Max
Gas Throughput	FLOW	MMscfd	76.0	91.6	122.0
Fuel Gas	FUEL	Btu/s	23,109.04	30,675.67	41,143.96
Suction Pressure	PSUC	psig	18.96	29.23	40.86
LPC Discharge Pressure	PDIS	psig	111.71	166.20	249.98
LPC Compression Ratio	PRAT		5.68	10.92	30.21
Combustion Air Inlet Temperature	CAIT	F	73.70	83.88	92.90
Combustion Air VIGV Servo Drive	IGV	%	8.99	11.13	13.84
GG HP Compressor Discharge Pressure	GGPD	psig	110.07	134.26	167.94
Compressor Load	LOAD	%	21.40	56.39	99.90
Power Turbine Speed	PTS	rpm	3,827.62	4,372.12	5,048.57
GG NL LP Compressor Speed	GGLP	rpm	5,353.33	5,764.43	5,973.33
GG NH HP Compressor Speed	GGHP	rpm	8,203.33	8,534.11	8,866.67
GG IP Turbine (Module 5) Temp	GGTE	F	554.80	613.97	675.60
Turbine Exhaust Temperature	EXHT	F	913.90	1,019.04	1,155.91
Fuel Gas Temperature	FGT	F	119.20	136.10	142.20
Temperature Outlet	TOUT	F	257.61	314.09	365.97
Engine Running Hours	RH	hours	1,409	32,971	39,545
Carbon Monoxide	CO	ppm	48.92	151.57	258.40
Carbon Dioxide	CO2	%	8.89	22.17	29.43
Oxygen	O2	%	13.16	14.96	19.31
Sulphur Dioxide	SO2	ppm	0.08	12.61	39.89
Nitric Oxide	NO	ppm	4.69	21.08	45.80

Two steps of correlation-specific measurement involve process data as input and measured emission data reading as an output variable, which is input to input variables and input to output variables.

In Fig. 5, the linear dependency between two random variables is measured using Pearson's correlation. The correlation table presents useful insight into the interrelationships among different performance metrics associated with a gas turbine engine.

It demonstrates that flow rate is correlated positively with emissions and fuel consumption, but negatively with efficiency metrics such as exhaust temperature. While the correlation between load and most other metrics is positive,

turnitin Page 115 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

the correlation between load and GGHP and FGT is relatively lower.

Flow and fuel consumption display an exceptionally robust positive correlation (0.97). This suggests that an increase in flow through the turbine corresponds to a proportionate increase in fuel consumption. This is probably due to the requirement that more fuel is required to produce additional power by consuming more gas.

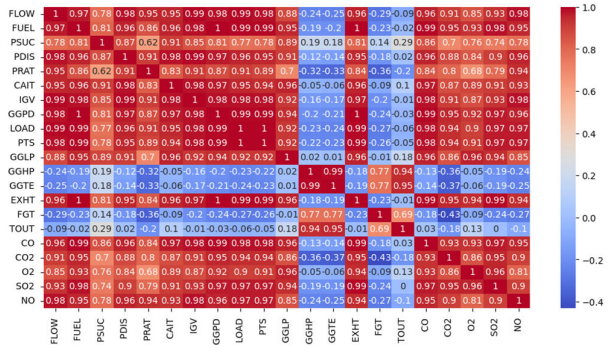


Fig. 5. Data correlation analysis

Fuel consumption and emissions demonstrate robust positive correlations, as evidenced by the coefficients of determination for NO (0.98), CO (0.95), CO_2 (0.91), and SO_2 (0.93). This means that emissions tend to increase in parallel with fuel consumption. This is because fuel combustion generates these pollutants.

Several efficiency metrics, including EXHT (-0.96) and TOUT (-0.09), show robust inverse correlations with flow. This indicates that an increase in turbine flow results in a related reduction in both the exhaust temperature (EXHT) and the turbine outlet temperature (TOUT). This indicates that higher flow rates result in more efficient operation of the engine.

Certain emission metrics and efficiency metrics exhibit robust inverse correlations. As an illustration, NO exhibits a negative correlation of -0.96 with EXHT and -0.18 with TOUT. This implies that there is a positive correlation between engine efficiency (as indicated by lower exhaust temperatures) and NO emissions, which are typically reduced.

IV. RESULT

A. GRNN

Generalized Regression Neural Network (GRNN) is well-suited for modelling the relationship between the anticipated output variables' test value and the actual test value in the dataset. The data points exhibit a clustering pattern around the regression line, and the R-squared value is approaching 1. This indicates that the model can precisely forecast the test value of the output variable, based on the features used for the linear regression model.

The measured carbon monoxide (CO) levels typically exceed the anticipated CO levels. Consequently, the PEMS model provides a lower estimate of the CO concentration. There is a consistent correlation between higher real CO values and higher CO readings overall in Fig. 6. However, the projected values indicate more variance from the actual measurements if the CO contents are higher.

There are obvious deviations and clusters of points that deviate from the perfect prediction line, suggesting that the model routinely underpredicts or overpredicts in certain locations.

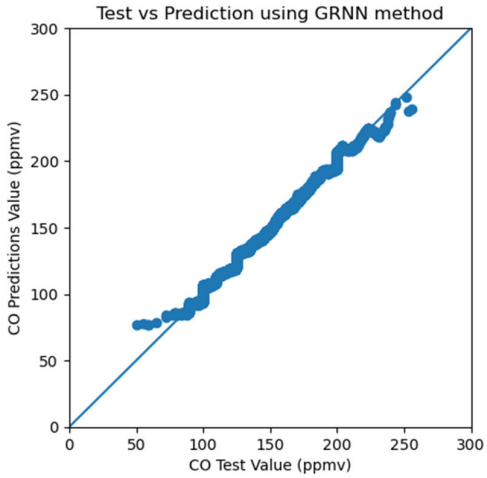


Fig. 6. CO test vs prediction using GRNN

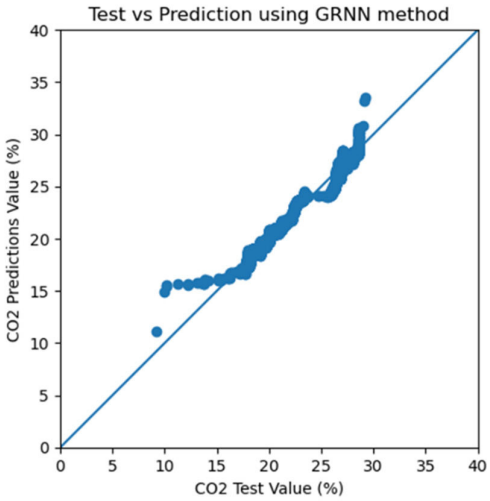


Fig. 7. CO_2 test vs prediction using GRNN

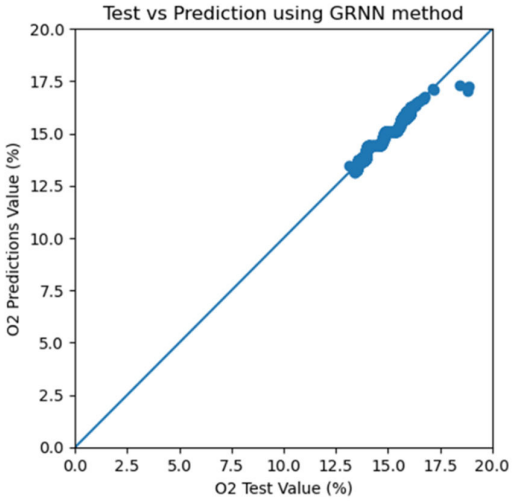


Fig. 8. O_2 test vs prediction using GRNN

turnitin Page 116 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

The data points in Fig. 6 are scattered around the diagonal line, which shows the model's predictions aren't perfect. But there's a clear trend that the higher of actual CO measurement, the higher of the predicted CO value, and vice versa for lower concentrations. This means the model does a good job of following the general trend of the CO levels, even if it's not always exactly right.

The graph in Fig. 7 shows a positive correlation between measured and predicted CO_2 concentration (ppmv) using the GRNN method, indicating the model predicts higher CO_2 levels when actual levels are high and vice versa, but with some scattered data points suggesting room for improvement in accuracy.

B. Decision Tree

The Fig. 9, Fig. 10, and Fig. 11 below represent the results of Decision Tree (DT) modelling and the correlation between the predicted test values of the output variables and the actual test values in the dataset.

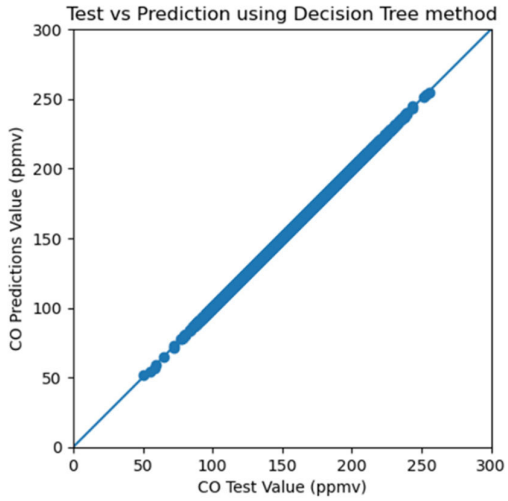


Fig. 9. CO test vs prediction using Decision Tree

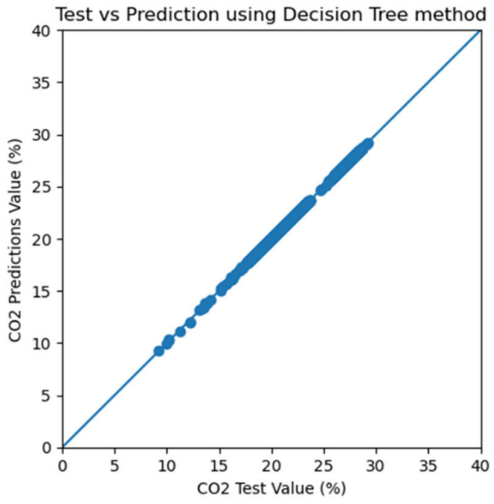


Fig. 10. CO_2 test vs prediction using Decision Tree

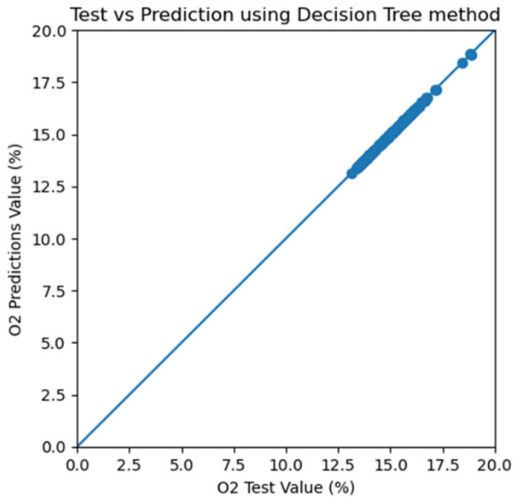


Fig. 11. O_2 test vs prediction using Decision Tree

The Decision Tree method demonstrates outstanding predictive performance for CO, CO_2, O_2, SO_2, and NO, as evidenced by the graph Fig. 9 analysis. Similar to the Random Forest graph, the data points form a straight line along the diagonal, indicating very high accuracy and a strong correlation between the test and predicted values.

This high level of precision suggests that the Decision Tree method is highly effective in predicting, with predictions closely aligning with the actual test values. Overall, the Decision Tree method's performance is comparable to that of the Random Forest method, highlighting its robustness and reliability for this task.

C. Random Forest

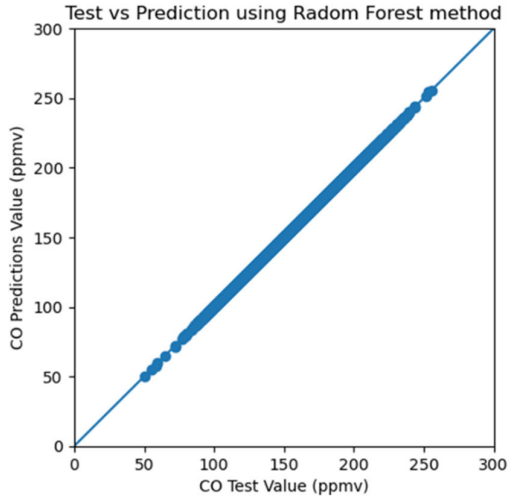


Fig. 12. CO test vs prediction using Random Forest

The graphics in Fig. 12, Fig. 13, and Fig. 14 illustrate the outcome of Random Forest (RF) modelling and the result of the correlation between the predicted test values of the output variables and the actual test values in the dataset. Fig. 12 indicates a very high correlation between the test and predicted values, suggesting that the Random Forest method has very high accuracy in predicting CO values. The model predictions are almost perfect, as shown by the close alignment of the points with the diagonal line.

turnitin Page 117 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

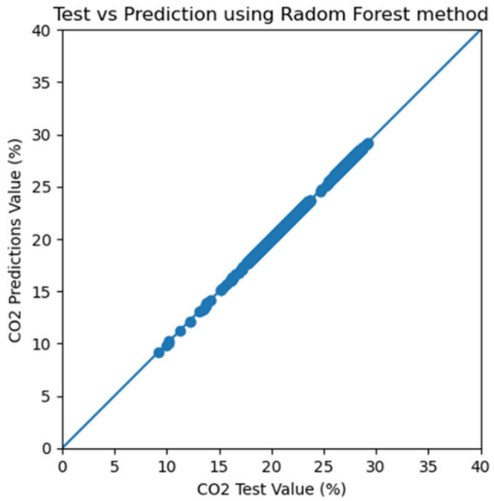


Fig. 13.CO_2 test vs prediction using Random Forest

The Random Forest and Decision Tree methods exhibit almost perfect accuracy in predicting, as evidenced by the tight clustering of points along the diagonal line in the graphs. This indicates a very strong correlation between the test and predicted values. In contrast, the GRNN method, while still accurate, shows more variation and less precision, particularly at higher values.

In terms of consistency, both the Random Forest and Decision Tree methods deliver very consistent results across the entire range of values. The GRNN method maintains good consistency at lower values but demonstrates less reliability at higher values.

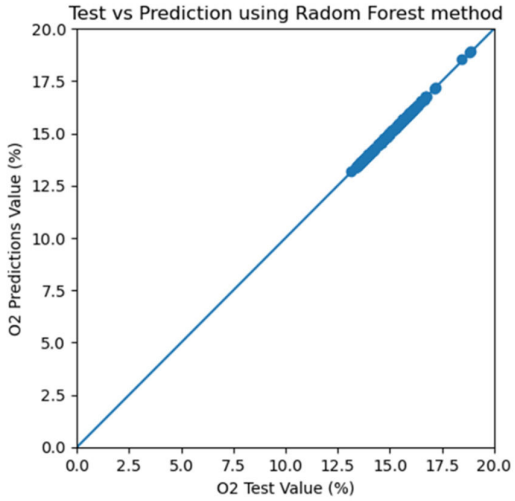


Fig. 14. O_2 test vs prediction using Random Forest

Based on this graphical analysis, the Random Forest and Decision Tree methods are preferable for prediction due to their superior accuracy and consistency. The GRNN method, while useful, may require further tuning or additional data to enhance its predictions, particularly for higher values.

D. Model Performance Evaluation

The Mean Squared Error (MSE) is a metric that quantifies the average of the squared differences between the projected values and the actual values, a lower mean squared error (MSE) value suggests superior performance.

The Mean Absolute Error (MAE) is a metric that quantifies the average absolute difference between the anticipated values and the actual values, a lower mean absolute error (MAE) value suggests superior performance.

The MAPE (Mean Absolute Percentage Error) is a metric that calculates the average absolute percentage difference between the projected values and the actual values, a lower Mean Absolute Percentage Error (MAPE) signifies superior performance.

R^2 indicates the amount of variation in the dependent variable that can be accounted for by the independent variables, a higher R^2 value suggests a stronger correlation between the model and the data.

TABLE III. MODEL PERFORMANCE INDICATOR

		GRNN	DT	RF
CO	MSE	5.64171	0.00114	0.00057
	MAE	1.77527	0.00600	0.00514
	MAPE	0.01258	0.00005	0.00004
	R^2	0.99475	1.00000	1.00000
CO_2	MSE	0.20682	0.00010	0.00003
	MAE	0.32500	0.00059	0.00059
	MAPE	0.01448	0.00003	0.00003
	R^2	0.98440	1.00000	1.00000
O_2	MSE	0.01012	0.00000	0.00000
	MAE	0.06979	0.00013	0.00015
	MAPE	0.00470	0.00001	0.00001
	R^2	0.98950	1.00000	1.00000
SO_2	MSE	0.15401	0.00005	0.00002
	MAE	0.27936	0.00091	0.00113
	MAPE	0.17964	0.00014	0.00022
	R^2	0.99853	1.00000	1.00000
NO	MSE	0.28317	0.00004	0.00004
	MAE	0.36188	0.00116	0.00124
	MAPE	0.03444	0.00006	0.00006
	R^2	0.99361	1.00000	1.00000

The TABLE III. above presents a comparison of the performance of three distinct machine learning models: Generalized Regression Neural Network (GRNN), Decision Tree (DT), and Random Forest (RF), to several target output variables (CO, CO_2, O_2, SO_2, and NO).

According to the R^2 values, all models appear to be doing exceptionally well, with values approaching 1.0 for the majority of target variables.

The Random Forest (RF) algorithm demonstrates superior performance across all metrics, consistently earning the lowest Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) values for the majority of goals. The Decision Tree (DT) algorithm also has strong performance, with MSE and MAE values that are highly similar to those of the Random Forest (RF) algorithm in the majority of cases.

Generalized Regression Neural Network (GRNN) exhibits the lowest performance as indicated by its MSE and MAE values, although maintains high R^2 values. However, the selection of the optimal model might be contingent on the relative importance of minimizing absolute errors versus squared errors for the given application.

E. Relatives Variables Importances

TABLE IV. RELATIVE VARIABLE IMPORTANCES

	CO	CO_2	O_2	SO_2	NO
FLOW	5.61E-04	3.02E-05	7.97E-04	7.93E-05	8.91E-05
FUEL	6.38E-01	8.78E-01	1.25E-01	7.98E-01	7.29E-01
PSUC	2.57E-05	4.43E-05	4.17E-05	2.64E-06	2.24E-05
PDIS	4.39E-05	1.86E-04	5.94E-04	2.22E-05	9.65E-03
PRAT	7.09E-03	1.24E-03	6.43E-04	9.70E-02	1.64E-01
CAIT	3.45E-04	6.04E-03	2.66E-04	5.86E-03	2.29E-05
IGV	2.57E-04	4.12E-05	1.10E-03	4.76E-05	7.01E-04
GGPD	1.96E-01	1.34E-03	4.80E-02	2.54E-03	5.29E-03
LOAD	7.11E-02	3.81E-03	9.44E-04	7.78E-03	2.64E-03
PTS	1.29E-03	9.11E-02	1.10E-04	8.49E-02	6.83E-02
GGLP	7.06E-02	1.71E-04	1.05E-04	1.60E-05	1.85E-04
GGHP	2.07E-07	5.19E-07	6.80E-07	5.08E-07	1.48E-06
GGTE	2.87E-07	3.12E-07	9.05E-07	2.02E-08	6.74E-07
EXHT	1.45E-02	1.85E-02	8.23E-01	3.71E-03	2.04E-02
FGT	3.19E-07	2.23E-07	1.50E-06	6.63E-08	4.70E-07
TOUT	2.10E-07	3.88E-07	7.95E-07	2.08E-08	1.66E-07

The Relative Variable Importance (RVI) metric is employed to quantify the significance of each feature (variable) in prediction. RVI assesses the contribution of each feature to the model's predictions by assigning it a score. Preference is given to characteristics that have higher RVI scores over those that have lower scores.

According to the modelling outcome, FUEL is the primary input variable that significantly influences the prediction of CO, CO_2, SO_2, and NO emissions. Consequently, the input variables TOUT, FGT, GGHP, GGTE, and TOUT will have the least influence.

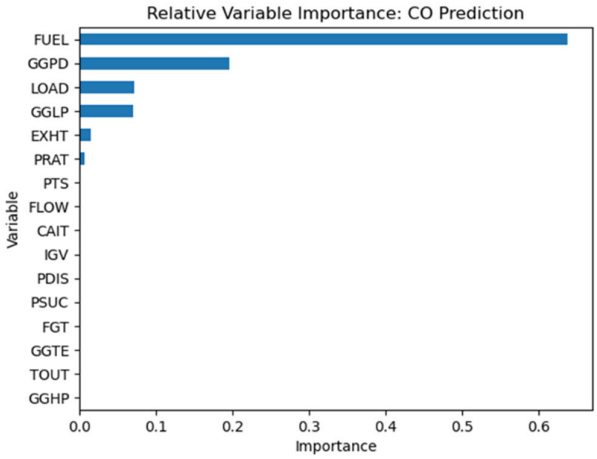


Fig. 15. RVI of CO prediction

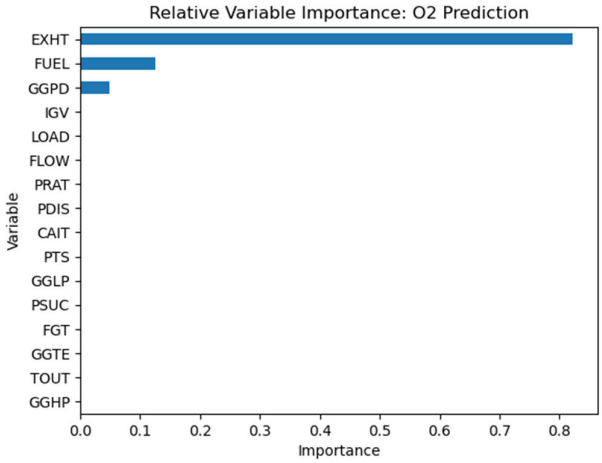


Fig. 16. RVI of O_2 prediction

F. Conclusions

Random Forest (RF) demonstrates superior overall prediction performance. The MSE, MAE, and MAPE values are at their minimum, suggesting that the predictions closely align with the actual values in terms of squared errors, absolute errors, and percentage errors.

The Decision Tree (DT) algorithm likewise has strong performance, as indicated by its Mean Squared Error (MSE) and Mean Absolute Error (MAE) values, which are very similar to those of the Random Forest (RF) algorithm. This indicates that the decision tree effectively captures the significant elements of the data for forecasting.

Generalized Regression Neural Network (GRNN) exhibits the poorest performance when evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). Although the R^2 value is high, suggesting a strong match, the elevated MSE and MAE indicate that the prediction errors might be bigger in comparison to DT and RF.

G. Recommendations

Determining the optimal model selection is dependent on specific requirements. The following considerations must be considered:

- When it comes to minimizing errors in predicting values, Random Forest is the best choice due to its lowest Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).
- Model interpretability is crucial when it comes to understanding the reasoning behind a model's predictions. In this regard, Decision Trees are often more straightforward to interpret than Random Forests.
- Random Forest typically necessitates a larger amount of data to attain optimal performance, in comparison to Decision Tree or Generalized Regression Neural Network. If the available data is restricted, it would be more advantageous to use either a Decision Tree or Generalized Regression Neural Network for analysis.

REFERENCES

[1] O. Kochueva and K. Nikolskii, "Data analysis and symbolic regression models for predicting CO and NOx emissions from gas turbines," *Computation*,

vol. 9, no. 12, Dec. 2021, doi: 10.3390/computation9120139.

[2] W. S. Y. Hung and F. Langenbacher, "PEMS: Monitoring NOx Emissions from Gas Turbines," 1995. [Online]. Available: http://journals.asmedigitalcollection.asme.org/GT/proceedings-pdf/GT1995/78828/V005T15A016/2406819/v005t15a016-95-gt-415.pdf

[3] H. Kaya, P. Tüfekci, and E. Uzun, "Predicting CO and NOx emissions from gas turbines: Novel data and a benchmark PEMS," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 6, pp. 4783–4796, 2019, doi: 10.3906/ELK-1807-87.

[4] K. K. Botros, C. Williams-Gossen, S. Makwana, and L. Siarkowski, "Presented at the 19 th Symposium on Industrial Application of Gas Turbines (IAGT) Data from Predictive Emission Modules Implemented on GE-LM1600, GE-LM2500 and RR-RB211Gas Turbines Employed in Natural Gas Compressor Stations."

[5] M. Si, T. J. Tarnoczi, B. M. Wiens, and K. Du, "Development of Predictive Emissions Monitoring System Using Open-Source Machine Learning Library-Keras: A Case Study on a Cogeneration Unit," *IEEE Access*, vol. 7, pp. 113463–113475, 2019, doi: 10.1109/ACCESS.2019.2930555.

[6] N. Bonavita, G. Ciarlo, and A. SpA, "Inferential Sensors for Emission Monitoring: An Industrial Perspective," 2014. [Online]. Available: www.seipub.org/fiee

[7] K. Smith and D. Cole, "Software versus hardware approach to emissions monitoring," *IEEE Trans Ind Appl*, vol. 53, no. 2, pp. 1717–1721, Mar. 2017, doi: 10.1109/TIA.2016.2639456.

[8] D. F. Specht, "A general regression neural network," *IEEE Trans Neural Netw*, vol. 2, no. 6, pp. 568–576, 1991, doi: 10.1109/72.97934.

[9] I. Izonin, R. Tkachenko, M. Gregus ml, K. Zub, and P. Tkachenko, "A GRNN-based approach towards prediction from small datasets in medical application," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 242–249. doi: 10.1016/j.procs.2021.03.033.

[10] S. A. Kalogirou, "Artificial intelligence for the modeling and control of combustion processes: A review," 2003. doi: 10.1016/S0360-1285(03)00058-3.

[11] F. S. Alakbari, M. E. Mohyaldinn, M. A. Ayoub, A. A. Salih, and A. H. Abbas, "A decision tree model for accurate prediction of sand erosion in elbow geometry," *Heliyon*, vol. 9, no. 7, Jul. 2023, doi: 10.1016/j.heliyon.2023.e17639.

[12] T. A. Munshi, L. N. Jahan, M. F. Howladar, and M. Hashan, "Prediction of gross calorific value from coal analysis using decision tree-based bagging and boosting techniques," *Heliyon*, vol. 10, no. 1, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23395.

[13] A. Arifuddin, G. S. Buana, R. A. Vinarti, and A. Djunaidy, "Performance Comparison of Decision Tree and Support Vector Machine Algorithms for Heart Failure Prediction," *Procedia Comput Sci*, vol. 234, pp. 628–636, 2024, doi: 10.1016/j.procs.2024.03.048.

[14] B. K. Meher, M. Singh, R. Birau, and A. Anand, "Forecasting stock prices of fintech companies of India using random forest with high-frequency data," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 1, Mar. 2024, doi: 10.1016/j.joitmc.2023.100180.

[15] X. Zhang *et al.*, "Improved random forest algorithms for increasing the accuracy of forest aboveground biomass estimation using Sentinel-2 imagery," *Ecol Indic*, vol. 159, Feb. 2024, doi: 10.1016/j.ecolind.2024.111752.

[16] Y. Li *et al.*, "Forecasting Monthly Water Deficit Based on Multi-Variable Linear Regression and Random Forest Models," *Water (Switzerland)*, vol. 15, no. 6, Mar. 2023, doi: 10.3390/w15061075.

[17] L. Breiman, "Random Forests," 2001.

[18] A. Jierula, S. Wang, T. M. Oh, and P. Wang, "Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data," *Applied Sciences (Switzerland)*, vol. 11, no. 5, pp. 1–21, Mar. 2021, doi: 10.3390/app11052314.

turnitin Page 120 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Hybrid PSO-CNN Model for Cross-Domain Adaptation Sentiment Analysis

1st Ummu Fatihah Mohd Bahrin
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA (UiTM)
Kuala Terengganu, Malaysia
ummufatihah@uitm.edu.my

2nd Hamidah Jantan
College of Computing, Informatics and Mathematics
Universiti Teknologi MARA (UiTM)
Kuala Terengganu, Malaysia
hamidahjtn@uitm.edu.my

***Abstract*—Sentiment analysis (SA) has garnered significant attention due to its application in understanding and interpreting human emotions from text data. Traditional SA models often face challenges when applied across different domains because of varying vocabulary, context, and language usage. Cross-domain SA aims to bridge this gap by transferring knowledge from one domain to another. Data distributions are different across domains and a possible solution is to learn a different system for each domain. However, it is challenging to design a resilient and cost-effective sentiment classifier. CNN has proven effective for text classification tasks, including SA. This paper explores the development and application of hybrid PSO-CNN models for cross-domain SA. The hybrid PSO-CNN model is designed to learn transferable features from source domains to target domains. Hyperparameter tuning is carefully performed to achieve optimal model performance in different domains. The hybrid PSO-CNN model demonstrates significant improvements in hyperparameter tuning and overall performance in the target domain. The model achieves a high accuracy of about 98.5% for the domain Book to DVD. The findings highlight the effectiveness of the hybrid PSO-CNN model in cross-domain sentiment analysis. Future research may explore advanced domain adaptation techniques and additional bio-inspired algorithms to further enhance model performance.**

Keywords— bio-inspired, cross-domain, domain adaptation, PSO-CNN, hybrid CNN, sentiment analysis

I. Introduction

Sentiment analysis (SA), a subfield of natural language processing (NLP), has gained significant attention due to its applications in understanding and interpreting human emotions from text data. Traditional SA models often face challenges when applied across different domains due to varying vocabulary, context, and language usage. Cross-domain sentiment analysis aims to bridge this gap by transferring knowledge from one domain to another [1]. In recent years, convolutional neural networks (CNNs) have proven effective for text classification tasks, including SA [2].

Tareq et.al [1] identify domain adaptation which is also referred to as cross-domain learning, as the involvement of annotated data from the source domain either independently or in combination with data from the target domain to build a model capable of predicting unannotated instances in the target domain. The biggest challenge in cross-domain SA is data labelling. The insufficiency of labelled data [3] and the high cost of data tagging create major challenges. Manual labelling is both time-consuming and expensive, especially in specific domains. However, leveraging strategies such as active learning [4], semi-supervised learning, distant supervision, and weak supervision [5] can help mitigate these challenges by reducing the dependence on manually labelled data.

Domain adaptation in SA refers to building models that can generalize well to new domains without additional labelled data. Data distributions are different across domains and a possible solution is to learn a different system for each domain. It is challenging to design a resilient and cost-effective sentiment classifier because each domain has a different vocabulary [6]. Furthermore, the challenge of domain adaptation is discrepancies in data distributions across different domains that make it difficult to directly apply models trained in one domain to another.

The insufficiency or absence of labelled data in the target domain further complicates adaptation efforts, constraining model adaptability [7]. Identifying domain-invariant features is difficult because certain domain-specific features might not translate well across different contexts, requiring complex strategies while adapting models [8]. Additionally, biases and unbalanced data distribution can continue from the source domain to other domains, affecting model performance [9].

The lack of guarantee that models trained in one domain will perform well in others, as noted by Rozie et.al [10] highlights the complexity of cross-domain SA. The most highlighted issues in cross-domain are data labelling, insufficiency of labelled data [11], domain shift, lack of uniformity, dataset dependency [12], and algorithm design. Therefore, it is crucial to use robust strategies to address these challenges.

SA using the deep learning method produces a high performance. CNN has achieved successful results [13]. Therefore, we are interested in exploring a new approach for cross-domain sentiment analysis using a hybrid Particle Swarm Optimization (PSO) method. This study explores the development and application of hybrid PSO-CNN models for cross-domain SA.

The subsequent sections of this paper are structured as follows: the next section provides the related work in cross-domain and the hybrid PSO-CNN algorithm in various domains. Following this, the subsequent section presents the methodology used in this study. Then after the explanation of the analysis process and a further discussion of performance in cross-domain SA techniques. Finally, the paper concludes with a summary section.

II. Related Work

A. Cross Domain Application in SA

Cross-domain SA aims to broaden the efficiency of SA models across various datasets and domains. Traditional SA

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

models are typically trained and assessed using data from a single domain or source, which may lack generalizability to new or different domains. Cross-domain SA addresses this challenge by adapting or transferring knowledge [14] acquired from one domain to another. Table 1 shows similar works in cross-domain SA applications.

TABLE I. SIMILAR WORKS IN CROSS-DOMAIN SENTIMENT ANALYSIS APPLICATIONS

Reference	Datasets	Algorithm / Technique	Performance
[1]	Amazon multidomain product datasets	BERT and bidirectional gated recurrent (BiGRU)	Comprehensive experiments showing the effectiveness of the proposed model over state-of-the-art techniques
[2]	online_shopping_10_cats dataset	CNN-BiLSTM-TE	Outperformed other models by a higher F1 score
[3]	Laptop (L), Restaurant (R), Device (D), and Service (S) datasets	Cross-Domain Review Generation (CDRG)	The baseline BERT model outperformed state-of-the-art domain adaptation methods
[4]	Office-31, Office-Home, COVID-19, Bing-Caltech, WebVision Datasets	Adversarial Network	The proposed approach outperformed existing methods

In a nutshell, most of the techniques reported above show great performance in cross-domain applications. Most scholars have endeavoured to address these challenges in cross-domain sentiment analysis using a variety of approaches. Researchers aim to improve the accuracy and applicability of SA in diverse real-world scenarios, covering the way for more effective decision-making and insights across various domains.

B. Hybrid Bio-inspired Algorithm with CNN

Bio-inspired algorithms such as neural networks, Ant Colony Algorithms (ACO), Particle Swarm Optimization (PSO), and others have been applied in almost every area of science [17], engineering, and business management [18]. Bio-inspired computing techniques have become increasingly significant in contemporary engineering research, especially for addressing optimization problems. These stochastic search methods aim to attain local optima solutions for large-scale optimization problems [19]. Conventional mathematical optimization approaches often struggle with local optima, which leads to the emergence of alternate derivative-free metaheuristic global optimization techniques. Because the traditional optimization methods often cannot escape local optima, new global optimization techniques that do not rely on metaheuristics have been developed to tackle this issue. [20] stated that examples of such techniques include genetic algorithms, particle swarm optimization, simulated annealing, etc. Tables 2 and 3 detail various studies on hybrid bio-inspired algorithms and hybrid CNN models in SA and text classification.

TABLE II. HYBRID BIO-INSPIRED SWARM ALGORITHM IN SA

Model	Dataset	Performance	Findings
SCO-FS-BPSO and SVM [5]	Hotel reviews are written in Chinese	93.84%	F-BPSO is better suited for feature selection in sentiment classification,
VPCNN-ABiLSTM with chimp (OBL-CHOA) [6]	Amazon, IMDB, Yelp, and Travel Dataset	96.50%	The feature selection technique was 6.9% higher than the existing BiLSTM-CNN
Hybrid Harris Hawks Optimization (HHO) [7]	Tweets related to COVID-19	96.73%	Comparison from different classifiers like Naïve Bayes, kNN, Random Forest, CNN-RNN, and AC-BiLSTM
GA with CNNRF [8]	IMDB	96.40%	The proposed method CNNRF with genetic algorithm hyperparameter tuning

TABLE III. HYBRID CNN ALGORITHM IN TEXT CLASSIFICATION

Model	Dataset	Performance	Findings
BiGRU-Att-HCNN **[9]**	Benchmark dataset - IMDB, Yelp2013 and TSB	92.82%	BiGRU-Att-HCNN effectively captures feature information at different text levels
Hybrid CNN-LSTM Model **[10]**	IMDB movie review dataset and Amazon movie reviews	91.00%	Outperformed both ML and DL models
BLSTM-CNN **[11]**	Thai Economy Twitter, Wisesight, ThaiTales	77.07%	Hybrids improve the overall performance.
Hybrid CNN-LSTM **[12]**	3000 tweets DKI Jakarta geocode	83%	CNN-LSTM Hybrid Model performs better with the words+emoji dataset
Hybrid CNN-RNN model **[13]**	ISOT dataset (fake news detection)	90%	Recommended considering more complex neural network architectures

TABLE IV. PSO-CNN APPLICATION IN VARIOUS DOMAINS

References	Algorithm / Technique	Dataset	Performance
[14]	Multi-level Particle Swarm Optimization (MPSO)	CIFAR-10, CIFAR-100, and MDRBI	MPSO-CNN perform significantly better
[15]	PSO-DNNs	MNIST, CIFAR-10	PSO improves the performance of existing architectures.
[16]	cPSO-CNN	CER	cPSO-CNN performs

			competitively when compared with several reported algorithms
[17]	PSO-CNN	Hindi speech dataset	The proposed CNN architecture performs similarly to other state-of-the-art methods

The fusion of bio-inspired optimization techniques such as PSO with CNN [16] holds considerable promise for advancing classification accuracy and overall performance in various text classification tasks. Table 4 shows the application of PSO-CNN in various domains. PSO applied to CNN proves to be versatile and effective across various domains, consistently improving model performance and demonstrating competitiveness with existing methodologies. These findings highlight PSO's potential in optimizing deep learning models for diverse applications, from image classification to speech recognition. All the studies represent a promising approach, despite the limited exploration of hybrid CNN with PSO in text classification.

Therefore, this study aims to explore the potential of a hybrid approach in optimizing text classification accuracy and performance through comprehensive empirical evaluation and comparative analysis.

III. METHODOLOGY

A. Dataset Preparation

The process begins with the collection of a representative dataset that serves as the benchmark and covers samples from various domains. The dataset introduced by Blitzer et al. [34] contains a few domains, such as books, DVDs, electronics, and kitchen. In this study, we only used text reviews and the selected domains are books and DVDs for performing cross-domain sentiment classification. Both the Book and DVD domains have 1000 positive reviews and 1000 negative reviews, which are balanced datasets for analysis. The main reason for using only two domains is that the challenge lies in the limited time available to execute experiments, due to their exhaustive nature [18].

Once the dataset had been confirmed, we prepared text data by cleaning it of HTML tags, and non-alphabetic characters, converting it to lowercase, and making sure there were no extraneous whitespace characters. This preprocessing was necessary before performing tasks such as SA, text classification, or any other natural language processing tasks where clean text data was required. After preprocessing, we proceeded to tokenize the pre-processed text using a Tokenizer. Initially, the Tokenizer was fitted on the reviews from the Book dataset to build a vocabulary of words. Then, we applied the same Tokenizer to tokenize the reviews from the DVD dataset. This process involved converting each review into sequences of integers based on the Tokenizer's vocabulary. Both the DVD and book review sequences were padded to a maximum length of 100 tokens, with padding applied at the end of each sequence. This step was essential for feeding the data into machine learning models – all sequences had consistent dimensions.

Finally, we extracted the sentiment labels from the respective datasets. These labels represented the sentiment classification for each review, which served as the target variable for training and evaluating SA models. Overall, these preprocessing and tokenization steps were undertaken to prepare the text data for subsequent machine learning tasks, ensuring the data was structured and formatted appropriately for analysis.

B. PSO-CNN Model Analysis

A CNN consists of convolutional layers for feature extraction, which apply convolutional operations to the input data using filters to detect various features such as edges, textures, and patterns. Pooling layers then reduce the spatial dimensions of these feature maps, decreasing computational complexity and providing some translation invariance. Finally, fully connected layers take the flattened output from the previous layers and perform the final classification based on the extracted features. Table 5 describes the CNN architecture used in this study.

TABLE V. CNN ARCHITECTURE

Layer	Type	Details
Input Layer	Embedding	Embedding layer with input_dim=vocab_size, output_dim=embedding_dim, input_length=maxlen
Convolutional Layer	Conv1D	filters=best_params[0], kernel_size=best_params[1], activation='relu', kernel_regularizer=l2(0.001)
Pooling Layer	MaxPooling1D	pool_size=2
Flatten Layer	Flatten	Flattens the input
Dropout Layer	Dropout	dropout_rate=best_params[2]
Dense Layer	Dense	units=best_params[3], activation='relu', kernel_regularizer=l2(0.001)
Output Layer	Dense	units=1, activation='sigmoid'
Optimizer	Adam	learning_rate=best_params[4]
Loss	Binary Crossentropy	Binary crossentropy loss function
Metrics	Accuracy	Accuracy as the evaluation metric

Meanwhile, the PSO-CNN algorithm can be understood through its components and their interactions. PSO optimizes CNN hyperparameters, including filter weights and biases, learning rates and regularization parameters, and architecture parameters such as the number of layers and neurons per layer. The hybrid PSO-CNN algorithm aims to improve CNN's performance in tasks such as text sentiment analysis by optimizing these parameters more effectively than traditional optimization methods. This combination leverages PSO's ability to explore the solution space globally and CNN's capability to extract intricate features from data, thereby improving the overall effectiveness of the model for complex optimization problems in machine learning tasks. Figure 1 shows the flowchart of the CNN optimization process using PSO, and on the other hand, Table 6 shows the PSO hyperparameter setting taken in this study.

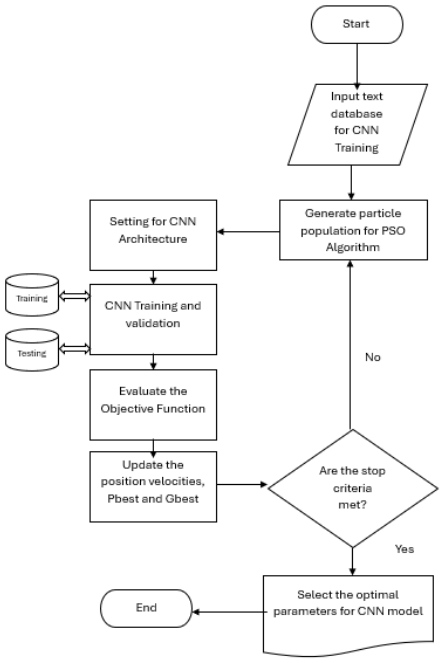


FIG. 1 FLOWCHART OF THE CNN OPTIMIZATION PROCESS USING PSO

TABLE VI. PARTICLE SWARM OPTIMIZATION (PSO) HYPERPARAMETERS

Hyperparameter	Description	Bounds
Filters	Number of convolutional filters	[32, 128]
Kernel Size	Size of the convolutional kernel	[3, 5]
Dropout Rate	Dropout rate for regularization	[0.1, 0.4]
Dense Units	Number of units in the dense layer	[10, 50]
Learning Rate	Learning rate for the optimizer	[0.0001, 0.001]
Swarm Size	Number of particles in the swarm	5
Max Iterations	Maximum number of iterations for PSO	10

C. *Model Evaluation*

The dataset is split into training and validation sets using a percentage split. training set of 80% and validation set of 20% data is used for validating the model during training to tune hyperparameters and prevent overfitting. Model valuation is crucial for assessing the performance of SA models across different domains. It involves assessing key metrics such as accuracy, precision, recall, F1-score, and AUC-ROC, using formulas that compare predictions against ground truth labels. These evaluation metrics help in understanding how well the SA model performs across different domains. Table 7 shows the experiment setup where the application of both models in different cross-domain scenarios, comparing their effectiveness in transferring knowledge between the Book and DVD domains.

TABLE VII. TABLE TYPE STYLES

Experiment	Cross Domain	Model
E1	Book → DVD	CNN
E2	DVD → Book	
E3	Book → DVD	PSO-CNN
E4	DVD → Book	

IV. RESULT AND DISCUSSION

Table 8 presents a comprehensive summary of performance metrics across four experiments. E3 demonstrates the highest accuracy among all experiments, achieving 98.50%, closely followed by E2 at 96.45% and E1 at 95.65%. These experiments consistently show high precision, recall, and F1 scores, reflecting their robust performance in correctly classifying instances. Notably, E2 and E3 achieve perfect AUC-ROC scores of 1.00, indicating exceptional discrimination ability between positive and negative classes. Overall, the results highlight the effectiveness of the models across various metrics, which means their suitability for applications requiring reliable classification in sentiment analysis tasks.

TABLE VIII. PERFORMANCE METRICS SUMMARY

Experiment	Accuracy	Precision	Recall	F1 Score	AUC - ROC
E1	95.65%	0.97	0.94	**0.96**	0.99
E2	96.45%	**0.98**	0.95	**0.96**	**1.00**
E3	**98.50%**	**0.98**	0.95	**0.96**	**1.00**
E4	94.60%	0.94	**0.96**	0.95	0.99

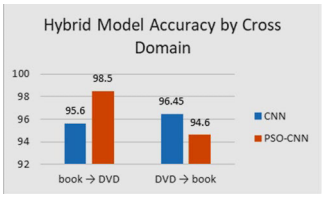


FIG. 2 MODEL PERFORMANCE BASED ON ACCURACY

Figure 2 shows the bar graph indicating the overall performance of the models by cross-domain based on accuracy. Accuracy provides a straightforward measure of the model's correctness in its predictions across all classes. Higher accuracy values generally indicate better overall model performance in classification tasks.

The log output is provided by the Particle Swarm Optimization (PSO) process using the pyswarms library. During the initialization and parameters, the configured to run for 10 iterations with the following parameters: a cognitive coefficient (c1) of 0.5, a social coefficient (c2) of 0.3, and an inertia weight (w) of 0.9. When the progress and completion, it shows that the best cost (objective function value) found during the optimization process is -1.0. The best position (set of parameters) found during the optimization process is:

- 1.33813386e+02 (approximately 133.81)
- 3.77799304e+00 (approximately 3.78)
- 2.65415747e-01 (approximately 0.27)
- 9.49697628e-03 (approximately 0.0095)

The PSO algorithm was run for 10 iterations using specified parameters, and it found the best set of parameters that resulted in a cost (objective function value) of -1.0. The best parameters found are approximately 133.81, 3.78, 0.27, and 0.0095.

There are some reasons why hybrid PSO-CNN (E3 – Book→DVD) has improved the hyperparameter tuning, which leads to better performance on the target domain.

PSO can significantly enhance the performance of CNN models, likely by improving their ability to capture complex patterns and features that generalize well across domains. Reviews of both books and DVDs are reviewed with consistent feature representations using similar language and sentiment expressions. This similarity aids the models, especially PSO-CNN, to learn transferable features during training, resulting in high cross-domain performance. [19] emphasize the importance of selecting instances from a source domain that are most like the target domain. The third reason is overfitting concerns as mentioned by [20, 21]. While PSO-CNN shows high performance in the Book→DVD transfer (E3), the slightly lower performance in the DVD→Book (E4) scenario might be due to the model being slightly overfitted to the source domain (DVD reviews), affecting its generalization to the target domain (Book reviews).

V. CONCLUSION

This study explores a hybrid PSO-CNN model for cross-domain sentiment analysis, addressing challenges of varied vocabulary and context across domains. Results indicate significant improvements in hyperparameter tuning, enhancing performance in the target domain with high accuracy, precision, recall, F1 scores, and AUC-ROC values. The study highlighted selecting source domain instances closely matching the target domain to mitigate overfitting concerns. Hybrid PSO-CNN models prove effective for diverse SA applications, suggesting potential for advanced domain adaptation techniques and integration of bio-inspired algorithms to further enhance model performance across domains. Future work could explore more advanced domain adaptation techniques and the integration of additional bio-inspired algorithms such as ACO, GA, etc.

ACKNOWLEDGEMENT

The registration fee is funded by Pembiayaan Yuran Prosiding Berindeks (PYPB), Tabung Dana Kecemerlangan Pendidikan (DKP), Universiti Teknologi MARA (UiTM), Malaysia.

REFERENCES

[1] R. K. Singh, M. K. Sachan, and R. B. Patel, "Cross-domain opinion classification via aspect analysis and attention sharing mechanism," Concurrency and Computation: Practice and Experience, vol. 34, no. 15, p. e6957, 2022, doi: https://doi.org/10.1002/cpe.6957.

[2] Y. Y. Zeng, R. R. Zhang, L. Yang, and S. J. Song, "Cross-Domain Text Sentiment Classification Method Based on the CNN-BiLSTM-TE Model," (in English), Journal of Information Processing Systems, Article vol. 17, no. 4, pp. 818-833, Aug 2021, doi: 10.3745/jips.04.0221.

[3] J. Yu, C. Gong, and R. Xia, "Cross-domain review generation for aspect-based sentiment analysis," in Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 4767-4777.

[4] Z. Han, X. J. Gui, H. Sun, Y. Yin, and S. Li, "Towards Accurate and Robust Domain Adaptation Under Multiple Noisy Environments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 5, pp. 6460-6479, 2023, doi: 10.1109/TPAMI.2022.3215150.

[5] L. Shang, Z. Zhou, and X. Liu, "Particle swarm optimization-based feature selection in sentiment classification," Soft Computing, vol. 20, no. 10, pp. 3821-3834, 2016/10/01 2016, doi: 10.1007/s00500-016-2093-2.

[6] D. A. J. Daniel and M. J. Meena, "Deep learning-based hybrid sentiment analysis with feature selection using optimization algorithm," Multimedia Tools and Applications, pp. 1-24, 2023.

[7] R. Seth and A. Sharaff, "Sentiment Data Analysis for Detecting Social Sense after COVID-19 using Hybrid Optimization Method," SN Computer Science, vol. 4, no. 5, p. 568, 2023.

[8] C. Siji George, "Genetic Algorithm Based Hybrid Model Of Convolutional Neural Network And Random Forest Classifier For Sentiment Classification," Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 2, pp. 3216-3223, 2021.

[9] Q. Zhu, X. Jiang, and R. Ye, "Sentiment analysis of review text based on BiGRU-attention and hybrid CNN," IEEE Access, vol. 9, pp. 149077-149088, 2021.

[10] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," Multimedia Tools and Applications, vol. 78, pp. 26597-26613, 2019.

[11] K. Pasupa and T. Seneewong Na Ayutthaya, "Hybrid deep learning models for thai sentiment analysis," Cognitive Computation, pp. 1-27, 2022.

[12] S. F. Pane, J. Ramdan, A. G. Putrada, M. N. Fauzan, R. M. Awangga, and N. Alamsyah, "A Hybrid CNN-LSTM Model With Word-Emoji Embedding For Improving The Twitter Sentiment Analysis on Indonesia's PPKM Policy," in 2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2022: IEEE, pp. 51-56.

[13] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," International Journal of Information Management Data Insights, vol. 1, no. 1, p. 100007, 2021/04/01/ 2021, doi: https://doi.org/10.1016/j.jjimei.2020.100007.

[14] P. Singh, S. Chaudhury, and B. K. Panigrahi, "Hybrid MPSO-CNN: Multi-level Particle Swarm optimized hyperparameters of Convolutional Neural Network," Swarm and Evolutionary Computation, vol. 63, p. 100863, 2021/06/01/ 2021, doi: https://doi.org/10.1016/j.swevo.2021.100863.

[15] P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, and J. R. Pastor, "Particle swarm optimization for hyper-parameter selection in deep neural networks," presented at the Proceedings of the Genetic and Evolutionary Computation Conference, Berlin, Germany, 2017. [Online]. Available: https://doi.org/10.1145/3071178.3071208.

[16] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," Swarm and Evolutionary Computation, vol. 49, pp. 114-123, 2019.

[17] V. Passricha and R. K. Aggarwal, "PSO-based optimized CNN for Hindi ASR," International Journal of Speech Technology, vol. 22, no. 4, pp. 1123-1133, 2019/12/01 2019, doi: 10.1007/s10772-019-09652-3.

[18] J. Blitzer, S. Kakade, and D. Foster, "Domain Adaptation with Coupled Subspaces," presented at the Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2011. [Online]. Available: https://proceedings.mlr.press/v15/blitzer11a.html.

[19] R. Remus, "Domain Adaptation Using Domain Similarity- and Domain Complexity-Based Instance Selection for Cross-Domain Sentiment Analysis," 2012 IEEE 12th International Conference on Data Mining Workshops, pp. 717-723, 2012.

[20] M. Mancini, S. R. Bulò, B. Caputo, and E. Ricci, "Best Sources Forward: Domain Generalization through Source-Specific Nets," 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1353-1357, 2018.

[21] M. A. Sultan, A. Sil, and R. Florian, "Not to Overfit or Underfit the Source Domains? An Empirical Study of Domain Generalization in Question Answering," in Conference on Empirical Methods in Natural Language Processing, 2022.

turnitin Page 125 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Enhanced Pooling Technique in Convolutional Neural Networks Model for Classification of *Magnoliophyta* Plant DNA Barcodes

Lilibeth P. Coronel
Graduate Programs
Technological Institute of the Philippines, Quezon City, Philippines
Mindanao State University at Naawan,
Naawan, Misamis Oriental, Philippines
lilibeth.coronel@msunaawan.edu.ph

Arnel C. Fajardo
College of Computer Science, Information and Communication Technology
Isabela State University
Cauayan, Isabela, Philippines
acfajardo2011@gmail.com

Ruji P. Medina
Graduate Programs
Technological Institute of the Philippines – Quezon City
Quezon City, Philippines
ruji.medina@tip.edu.ph

***Abstract*—Identifying plant Deoxyribonucleic acid (DNA) barcodes has gained substantial importance for biodiversity conservation and understanding evolutionary relationships. However, classifying these species remains challenging due to the complex nature of the DNA sequences. Convolutional Neural Networks (CNNs) have proven effective for pattern recognition tasks in DNA sequence classification, however, standard pooling techniques often result in the loss of important information. This study introduced a modified CNN model with an enhanced pooling technique called Horizontal Sequence Pooling, designed to retain important features and improve accuracy. The method is applied to classify the *Magnoliophyta* plant DNA barcodes of 75 genera and evaluated against the CNN model with standard max pooling and average pooling techniques. Results show that the proposed technique achieved the highest accuracy of 0.9801, the lowest validation loss of 0.0393, superior Area Under Curve – Precision and Recall (AUC-PR) of 0.9979, and Matthews Correlation Coefficient (MCC) of 0.9814, respectively, outperformed standard pooling techniques. These results indicate that Horizontal Sequence Pooling effectively extracts relevant features from DNA sequences, enhancing the classification accuracy, precision, recall, and balance across both positive and negative classes demonstrating its robustness in handling imbalanced datasets.**

Keywords—Convolutional Neural Networks, horizontal pooling, classification, DNA barcode, Magnoliophyta plant.

I. Introduction

Plant Deoxyribonucleic acid (DNA) barcode identification has gained importance as a research area, mainly because of growing biodiversity loss that needs global attention. The accurate classification of these plants becomes highly significant for several reasons, including the conservation of threatened species and the improvement of natural resource protection [1]. Despite the recent advances in genomic sequencing technologies, the effective classification of plant species based on DNA sequences remains a challenge [2]. The sequences have high phylogenetic diversity and intricate patterns, leading to classification complexity. Thus, an improved and tailored feature extraction technique for classification tasks is essential.

Convolutional Neural Networks (CNNs) have recently become popular for analyzing genomic sequences, achieving state-of-the-art accuracy in various genomic classification applications. It can capture spatial dependencies and patterns in the first layer filters to obtain salient feature representations [3]. However, the efficiency of CNNs is highly dependent on how well they preserve important features during the pooling process. In the pooling process, the spatial dimension is reduced to produce smaller feature maps while keeping important features' dominant and exact position. However, standard pooling techniques, such as max pooling and average pooling often encounter loss of relevant information [4], affecting the model's ability to classify between closely related classes.

In deep learning, key metrics such as precision and recall (sensitivity) are frequently used to assess the model's effectiveness. Precision calculates the accuracy of positive predictions, whereas recall evaluates the model's ability to distinguish all relevant instances. Standard pooling methods, such as max pooling and average pooling could sacrifice these performance measures by discarding crucial spatial information. Thus, there is a need for enhanced pooling technique that can improve the model's ability to retain important features and ultimately result in better classification results.

In this study, a CNN model with an enhanced pooling technique called Horizontal Sequence Pooling tailored for the classification of *Magnoliophyta* plant DNA sequences is proposed. The method aims to preserve relevant features and improve the model's ability to distinguish between similar species. Pooling is done in the horizontal axis of the feature maps where sequences are paired by applying positional max and average operations. By refining the pooling method, we seek to achieve higher precision and recall, ultimately enhancing the overall performance and reliability of the classification model. This study also compares Horizontal Sequence Pooling with max pooling and average pooling in terms of validation loss, accuracy, specificity, F1-score, Area Under Curve – Precision and Recall (AUC-PR), and Matthews Correlation Coefficient (MCC), which effectively captures the performance of the model's accuracy across positive and negative classes, particularly in imbalanced dataset.

II. Background and Related Works

A. Plant DNA Barcodes

In 2003, DNA barcoding technology was introduced by [5] for accurate species identification. This method uses a short, standardized DNA sequence marker, known as a DNA barcode, to rapidly and accurately identify species [6]. Since its inception, DNA barcoding has been widely

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

adopted in various applications and is considered a significant advancement in taxonomy [7].

The Consortium for the Barcode of Life (CBOL) Plant Working Group recommends internal transcribed spacer (ITS) region for plant barcoding, given its utility in resolving species-level differences and supplementing ribulose-1,5-bisphosphate carboxylase/oxygenase large subunit (*rbcL*) and maturase K (*matK*) genes data for comprehensive taxonomic identification. *Magnoliophyta* commonly known as angiosperms or flowering plants, represent the largest and most diverse group of land plants, encompassing approximately 300,000 species. The evolutionary success of *Magnoliophyta* is attributed to their complex vascular systems, varied life forms, and ecological adaptability, allowing them to dominate terrestrial ecosystems across the globe [8].

B. Pooling Operations

In convolutional neural networks, pooling is essential for reducing the dimensionality of extracted features. This process involves subsampling the features to generate multiple feature maps with lower resolutions.

Max pooling (1) is the predominant pooling technique employed in CNNs [9]. This approach selects the maximum activation within the pooling region, disregarding all other units. It effectively reduces the spatial dimensions of a feature map and imparts translational invariance to the network [10].

$$f_{max} = \max_{i \in R_{max}} a_i \quad (1)$$

Where R_{max} represents the pooling region and $\{\alpha_1,\ldots, \alpha_{|Rmax|}\}$ denotes a set of activations. The study of [11] employed max pooling as a feature extraction method, resulting in high accuracy on the test data.

On the other hand, average pooling (2) computes the statistical mean of a cluster of neurons within the feature map [12]. In this approach, the input image is divided into several distinct rectangular regions. The mean pixel value within each of these regions is calculated and used to form the output. Mathematically, it is defined as follows:

$$f_{ave}(X) = \frac{1}{N}\sum_{i}^{N=1} x_i \quad (2)$$

The vector x represents activations from a set N of permutations within a rectangular area of an image or channel. This approach can significantly reduce the distinctiveness of convolutional features, especially when many activations in the pooling area are zero [13]. In the study conducted by [14] [15], average pooling was applied in the model, demonstrating notable improvements over existing technologies.

C. Convolutional Neural Networks

CNNs are a type of deep learning algorithm designed primarily for image recognition and classification tasks. Excels in handling large-scale data and extracting hierarchical features through local receptive fields, parameter sharing, and translation invariance, reducing parameters and improving robustness [16]. The architecture, including convolutional, pooling, and fully connected layers, allows them to effectively capture local and global patterns in data.

Moreover, CNNs have been successfully used in genomics, particularly for DNA sequence classification, by leveraging their ability to automatically extract features and learn from complex and high-dimensional raw sequence data. Key factors contributing to their effectiveness include their capacity for automatic feature learning, scalability to handle large genomic datasets efficiently, and demonstrated high accuracy in DNA sequence classification tasks, outperforming traditional machine learning methods [11] [17] [18] [19] [20].

Recent advancements in applying CNNs to plant DNA sequence classification have shown significant progress, highlighting their potential in understanding plant genetics. For instance, the Deep-BSC model for predicting DNA binding sites in Arabidopsis thaliana utilizes max-pooling for improved precision and accuracy [21]. Additionally, a study of [22] presents CNN_FunBar, a CNN-based approach for classifying fungal ITS sequences, achieving over 93% accuracy and outperforming traditional machine learning algorithms and existing fungal taxonomy prediction software.

III. MATERIALS AND METHODS

A. Data Pre-processing and Augmentation

This study utilized *Magnoliphyta* plant DNA sequences from the public database: "Barcode of Life Data (BOLD) System" (https://www.boldsystems.org/). A collection of 14,761 DNA barcode samples as shown in Fig. 1, each linking scientific names to (ITS) nucleotide (A, C, G, T) sequences ranging from 209~234 bases in length, covering 75 genus classes.

One-hot encoding was used to convert sequences into a numerical format, preserving the positional information of each nucleotide. To manage sequences of different lengths, padding was employed to achieve uniformity in input dimensions for the model. And a Synthetic Minority Oversampling Technique (SMOTE) algorithm [23] [24] was utilized to address the imbalance within the dataset. The algorithm delineates minority classes using a predefined threshold, thus, classes exhibiting fewer occurrences are categorized as minority classes.

First 5 rows:

	genus_name_ID	nucleotides
0	1	TATCATGTCGCCCCCACCCACCAATCCTCTAGAGGACGTGTTTTGT...
1	2	ACATCGTCACCCCTTCCTCCACACTTAACCAGTTAAATGTGAGGCA...
2	3	TCACCCCTCCTCCTACTTCGGTGGACGGGGTGGAATGTGACCTCCT...
3	4	CGCATCGCGTCGCCCCCCAACCCATCATTCCCTCGCGGGAGTCGAT...
4	5	ATCGCGTCGCCCCCCGCACGCCGCTAGGCGTCGCGTGGGGCGGATA...

Last 5 rows:

	genus_name_ID	nucleotides
14756	73	ATCTCGTCGCCACCCCCTCTCGCGGGGGCGGCGGAGACTGGCCTCC...
14757	54	ATCGCGTCGCCCCCATCACACCYCTTGACGGGGATGTTTGAATGGG...
14758	11	GCCCATCCACGCTCGGCGAACGCCCGATGCGGACAATGGCCCTCCG...
14759	31	AAAGCTTTGCCCCCCCGCCGCTCGCTCAGAGAGAGCTGATGCTCGG...
14760	41	CATCGCGTCGCTCCCACCAACCTTCCCTTTGGGATGATTGGTTTGG...

Fig. 1 *Magnoliphyta* plant DNA sequence dataset.

B. The Enhanced Pooling Technique

Max pooling is a common choice for pooling layers in CNN architectures due to its ability to efficiently extract the most salient features from the input data [25]. On the other hand, average pooling has also been utilized in state-of-the-art image classification models, as it can increase model

stability [26]. However, these pooling methods can lead to the loss of important information, potentially hindering the model's ability to differentiate between closely related classes. Max pooling focuses solely on the maximum features and disregards the rest of the pooling area, while average pooling considers all the features, which can reduce the contrast of the resulting feature map. To address these limitations, horizontal sequence pooling is introduced by generating a representation that considers the maximum occurrence of nucleotide pairs and retains the most significant signals within the window by applying average operation, thereby ensuring that important features are not lost.

The Horizontal Sequence Pooling (HSP) operation pools a feature map by defining W as a window of the sequence, represented as a matrix with dimension 4 x m, where 4 corresponds to the one-hot encoding dimensions for the nucleotides A, C, G, and T, and m is the size of the window. For a given window W, features a and b are derived as the maximum values from the selected positions within the window as shown in (3) and (4). The HSP is then calculated as the average of features a and b as shown in (5).

$$feature_a = max\left(W_0, W_2\right) \quad (3)$$

$$feature_b = max\left(W_1, W_3\right) \quad (4)$$

$$HSP = average\left(feature_a + feature_b\right) \quad (5)$$

C. CNN Model

The model was designed with two one-dimensional convolutional layers tailored for sequential data analysis, equipped with 64 and 128 filters, respectively, and a kernel size of 3. The Rectified Linear Unit (ReLU) [27] was utilized as the activation function. To enhance the training efficiency, batch normalization [28] was employed. A horizontal sequence pooling layer with both pool and stride sizes set at 4 was utilized. The network also includes a flattening layer followed by two dense layers with 256 and 128 neurons, respectively. To mitigate overfitting, dropout [29] techniques are incorporated. The final layer of the model, designed for output classification, utilizes a softmax [30] activation function. Fig. 2 in the study illustrates the data pre-processing steps and the CNN model's architecture.

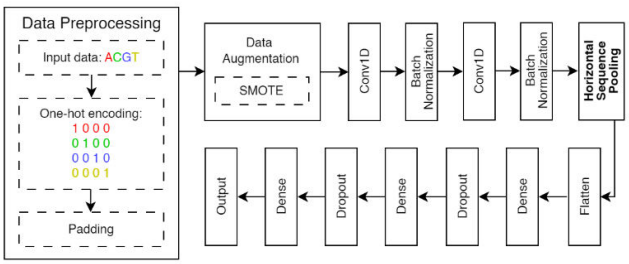


Fig. 2 CNN model with data pre-processing.

D. Performance Metrics

The performance of the proposed method was evaluated using the following metrics: Accuracy (6), Precision (7), Recall (8), Specificity (9), and F1-score (10). The AUC-PR and MCC functions of the TensorFlow are used to assess the trade-offs between precision and recall to provide a comprehensive score that reflects the model's performance across both positive and negative classes.

$$Accuracy = (TP + TN) / (Total\ no.\ of\ samples) \quad (6)$$

$$Precision = TP / (TP + FP) \quad (7)$$

$$Recall\ (Sensitivity) = TP / (TP + FN) \quad (8)$$

$$Specif icity = TN / (TN + FP) \quad (9)$$

$$F1 - score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (10)$$

IV. Experimental Findings and Results

The proposed model was experimented with using Python 3.9.12 on an Intel(R) Core i7 10870H CPU @ 2.20GHz with 32GB of random-access memory and 6GB NVIDIA GeForce RTX 3060 of graphics processing units. After implementing data augmentation techniques, the dataset expanded to a total of 40,738 instances. These instances were systematically allocated across training, validation, and testing subsets, comprising 70% (28,516), 10% (4,074), and 20% (8,148) of the dataset, respectively. Throughout the training process, the loss function utilized was categorical cross-entropy [31], and the optimization was managed using the Adam [32] optimizer set at a learning rate of 0.0001. Three models with a batch size of 16 are trained over 20 epochs, each employing a unique pooling technique— Horizontal Sequence Pooling, max pooling, and average pooling. Furthermore, Fig. 3 to 5 provide a detailed comparative convergence plot that illustrates the validation loss, accuracy, and AUC-PR for each pooling technique.

It is observed that there is a continuous decline of validation loss across the three pooling techniques as shown in Fig. 3. Among these, average pooling showed the highest validation loss at 0.207, demonstrating the least improvement and indicating its limited efficiency in feature extraction. In contrast, max pooling presented a slight improvement with a validation loss of 0.1601. However, Horizontal Sequence Pooling attained the most significant convergence compared to max pooling and average pooling, achieving the lowest validation loss of 0.0393.

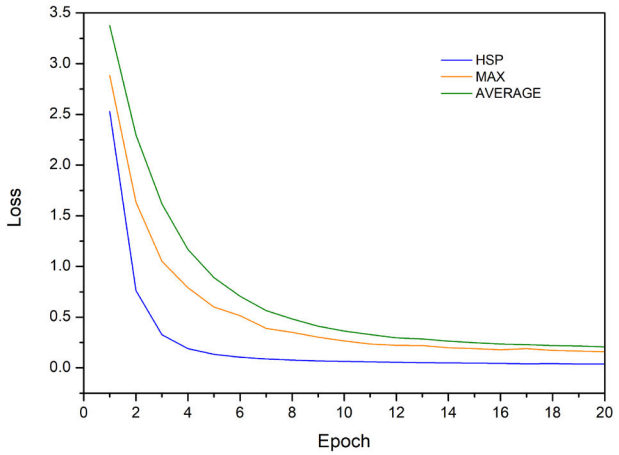


Fig. 3 Convergence plot of validation loss.

This superior performance suggests that Horizontal Sequence Pooling is highly effective in extracting relevant features and generalizes well on unseen data outperforming both max pooling and average pooling techniques.

As indicated in Fig. 4, the three pooling techniques show an increase in validation accuracy over time. However, notable differences in performance are evident. Horizontal Sequence Pooling has the highest accuracy at 0.9801, significantly outperforming max pooling and average pooling, which attain 0.9683 and 0.9431, respectively. These findings highlight the superior effectiveness of Horizontal Sequence Pooling in capturing intricate patterns and improving classification performance.

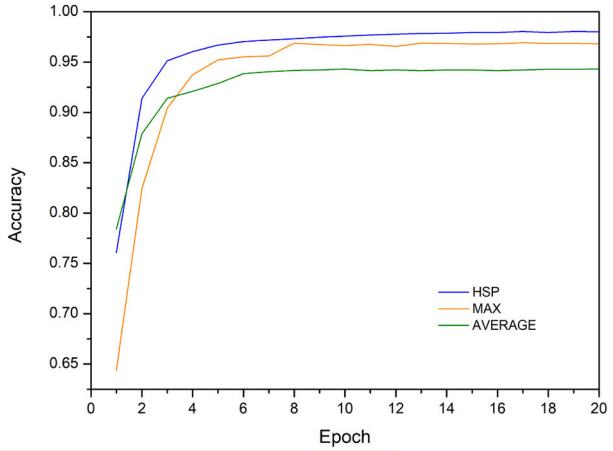


Fig. 4 Convergence plot of validation accuracy.

The learning curves for the AUC-PR during validation for the three pooling techniques are shown in Fig. 5. In comparison, max pooling and average pooling attained AUC-PR scores of 0.9915 and 0.9927, respectively. On the other hand, Horizontal Sequence Pooling demonstrated the most substantial performance, achieving the highest AUC-PR score of 0.9978. These results indicate that the model utilizing Horizontal Sequence Pooling is highly efficient in balancing accurate predictions and correctly identifying target sequences, outperforming the standard pooling methods in terms of precision-recall performance.

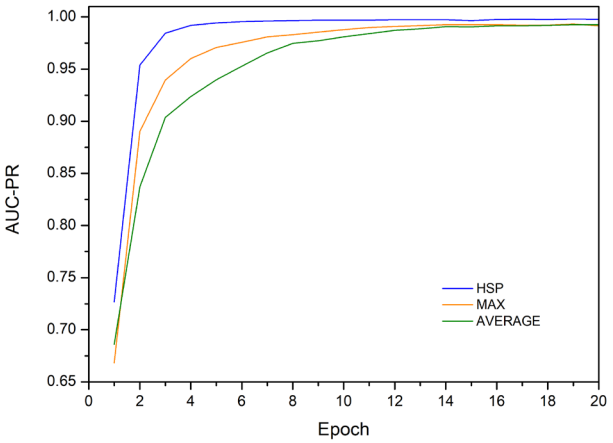


Fig. 5 Convergence plot of validation AUC-PR.

A comparative summary of the three CNN models with distinct pooling techniques evaluated in test datasets using multiple metrics is shown in Table 1.

The comparison of the accuracy result shows that Horizontal Sequence Pooling provides the highest accuracy compared to max pooling and average pooling, indicating superior performance in classifying plants under the phylum *Magnoliophyta*. Additionally, Horizontal Sequence Pooling consistently outperforms the other techniques in various metrics, notably achieving the highest AUC-PR score, which indicates superior discrimination and a well-balanced trade-off between precision and recall.

TABLE I. COMPARATIVE PERFORMANCE METRICS ACROSS DIFFERENT POOLING TECHNIQUES

Metric	Pooling Technique		
	Average Pooling	***Max Pooling***	***Horizontal Sequence Pooling***
Accuracy	0.9429	0.9688	0.9816
AUC-PR	0.9923	0.9921	0.9979
MCC	0.9433	0.9688	0.9814
Precision	0.9407	0.9682	0.9849
Recall (Sensitivity)	0.9506	0.9754	0.9873
Specificity	0.9992	0.9996	0.9997
F1-score	0.9398	0.9678	0.9842

In terms of sensitivity, Horizontal Sequence Pooling demonstrates its efficacy in accurately extracting positional features from the input data. Moreover, the MCC score implies the model's robust performance across both positive and negative classes. This indicates that the model with Horizontal Sequence Pooling excels in correctly identifying DNA sequences belonging to specific plant species among the 75 classes, as well as correctly identifying sequences that do not belong to the targeted species.

V. CONCLUSION AND FUTURE WORKS

This study introduced Horizontal Sequence Pooling as a new approach to address the limitations of max pooling and average pooling in the classification of plant DNA sequence under the *Magnoliophyta* phylum. The technique involves pooling of relevant features in horizontal specific positions. The technique excels as a highly effective feature extraction method that significantly improves the performance and reliability of the CNN model for DNA sequence classification. The results exhibit the superior performance of Horizontal Sequence Pooling in terms of reducing loss, increasing accuracy, AUC-PR, and MCC compared to standard pooling methods. This enhancement is expected to contribute significantly to the fields of bioinformatics and genomics, where accurate classification of large-scale genomic data is crucial.

Future work should focus on filtering the pooling technique and investigating its application across diverse genomic data to fully exploit its capabilities. Moreover, fine-tuning the parameters and configurations of the CNN model with HSP could lead to even better optimization for specific genomic classification tasks.

ACKNOWLEDGMENT

The authors acknowledge the financial support provided by the Philippine Government, assisted by the Commission on Higher Education (CHED) for the completion of this study.

REFERENCES

[1] Z. Hua, C. Jiang, S. Song, D. Tian, Z. Chen, Y. Jin, Y. Zhao, J. Zhou, Z. Zhang, L. Huang, and Y. Yuan, "Accurate identification of taxon-specific molecular markers in plants based on DNA signature sequence," *Molecular ecology resources,* vol. 23(1), pp. 106-117, 2023.

[2] L. Riza, M. Ammar, F. Rahman, Y. Prasetyo, M. Zain, H. Siregar, T. Hidayat, K. Fariza, A. Samah, and M. Rosyda, "Comparison of Machine Learning Algorithms for Species Family Classification using DNA Barcode," *Knowledge Engineering and Data Science*, 2023.

[3] P. Koo, and M. Ploenzke, "Improving representations of genomic sequence motifs in convolutional networks with exponential activations," *Nature machine intelligence,* vol. 3, pp. 258-266, 2021.

[4] P. Singh, P. Raj, and V. Namboodiri, "EDS pooling layer," *Image Vis. Comput.,* vol. 98, 2020.

[5] P. D. Hebert, A. Cywinska, S. L. Ball, and J. R. deWaard, "Biological identifications through DNA barcodes," *Proceedings. Biological sciences,* vol. 270(1512), pp. 313-321, 2003.

[6] P. D. N. Hebert, and T. R. Gregory, "The Promise of DNA Barcoding for Taxonomy," *Systematic Biology,* vol. 54, no. 5, pp. 852-859, 2005.

[7] R. T. Mampang, K. C. A. Auxtero, C. J. C. Caldito, J. M. Abanilla, G. A. G. Santos, and C. M. A. Caipang, "DNA Barcoding and Its Applications: A Review," *UTTAR PRADESH JOURNAL OF ZOOLOGY,* vol. 44, no. 20, pp. 69-78, 09/13, 2023.

[8] C. P. W. Group1, P. M. Hollingsworth, L. L. Forrest, J. L. Spouge, M. Hajibabaei, S. Ratnasingham, M. van der Bank, M. W. Chase, R. S. Cowan, D. L. Erickson, A. J. Fazekas, S. W. Graham, K. E. James, K.-J. Kim, W. J. Kress, H. Schneider, J. van AlphenStahl, S. C. H. Barrett, C. van den Berg, D. Bogarin, K. S. Burgess, K. M. Cameron, M. Carine, J. Chacón, A. Clark, J. J. Clarkson, F. Conrad, D. S. Devey, C. S. Ford, T. A. J. Hedderson, M. L. Hollingsworth, B. C. Husband, L. J. Kelly, P. R. Kesanakurti, J. S. Kim, Y.-D. Kim, R. Lahaye, H.-L. Lee, D. G. Long, S. Madriñán, O. Maurin, I. Meusnier, S. G. Newmaster, C.-W. Park, D. M. Percy, G. Petersen, J. E. Richardson, G. A. Salazar, V. Savolainen, O. Seberg, M. J. Wilkinson, D.-K. Yi, and D. P. Little, "A DNA barcode for land plants," *Proceedings of the National Academy of Sciences,* vol. 106, no. 31, pp. 12794-12797, 2009.

[9] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, pp. 111–118, 2010.

[10] I. V. P. D. Reyes, A. M. Sison, and R. P. Medina, "A Novel Fused Random Pooling Method for Convolutional Neural Network to Improve Image Classification Accuracy," *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pp. 1-5, 2019.

[11] H. Gunasekaran, K. Ramalakshmi, A. Arokiaraj, S. Kanmani, C. Venkatesan, and C. Dhas, "Analysis of DNA Sequence Classification Using CNN and Hybrid Models," *Computational and Mathematical Methods in Medicine*, 2021.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[13] W. Zhang, C. Li, G. Peng, Y. Chen, and Z. Zhang, "A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load," *Mechanical Systems and Signal Processing,* vol. 100, pp. 439-453, 2018.

[14] M. Shujaat, A. Wahab, H. Tayara, and K. T. Chong, "pcPromoter-CNN: A CNN-Based Prediction and Classification of Promoters," *Genes,* vol. 11, no. 12, pp. 15-29, 2020.

[15] F. Bieder, R. Sandkühler, and P. C. Cattin, "Comparison of Methods Generalizing Max- and Average-Pooling," *ArXiv,* vol. abs/2103.01746, 2021.

[16] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big data,* vol. 8(1), no. 53, 2021.

[17] C.-H. C. -H. Yang, K.-C. Wu, L.-Y. Chuang, and H.-W. Change, "DeepBarcoding: Deep Learning for Species Classification Using DNA Barcoding," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 19, no. 4, pp. 2158-2165, 2022.

[18] A. El-Tohamy, H. Maghwary, and N. Badr, "A Deep Learning Approach for Viral DNA Sequence Classification using Genetic Algorithm," *International Journal of Advanced Computer Science and Applications*, 2022.

[19] N. F. Soliman, S. M. Abd-Alhalem, W. El-Shafai, S. E. S. E. Abdulrahman, N. Ismaiel, E. S. M. El-Rabaie, A. D. Algarni, F. Algarni, A. A. Alhussan, and F. E. A. El-Samie, "Hybrid Approach for Taxonomic Classification Based on Deep Learning," *Intelligent Automation and Soft Computing,* vol. 32(3), pp. 1881-1891, 2022.

[20] R. Bandi, and T. Santhisri, "Implementation of a deep convolution neural network model for identifying and classifying Pleuropulmonary Blastoma on DNA sequences," *e-Prime - Advances in Electrical Engineering, Electronics and Energy,* vol. 5:100233, 2023.

[21] S. Bukhari Adnan Shah, A. Razzaq, J. Jabeen, S. Khan, and Z. Khan, "Deep-BSC: Predicting Raw DNA Binding Pattern in Arabidopsis Thaliana," *Current Bioinformatics,* vol. 16, no. 3, pp. 457-465, 2021.

[22] R. Das, A. Rai, and D. C. Mishra, "CNN_FunBar: Advanced Learning Technique for Fungi ITS Region Classification," *Genes,* vol. 14, no. 3, pp. 634, 2023.

[23] R. Blagus, and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics,* vol. 14, no. 1, 2013.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and P. W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[25] K. O'Shea, and R. Nash, "An Introduction to Convolutional Neural Networks," *CoRR,* vol. abs/1511.08458, 2015.

[26] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv*, 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* vol. 25, 2012.

[28] S. Ioffe, and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, pp. 448--456, 2015.

[29] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *ArXiv,* vol. abs/1207.0580, 2012.

[30] J. S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition," *Neurocomputing.* pp. 227-236, 1990.

[31] P. Murugan, "Implementation of Deep Convolutional Neural Network in Multi-class Categorical Image Classification," *ArXiv,* vol. 1801.01397, 2018.

[32] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

turnitin Page 130 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Flood susceptibility mapping using publicly available big data with Google Earth Engine and deep learning algorithms

Sackdavong MANGKHASEUM
Department of Electrical and Space Systems Engineering
Kyushu Institute of Technology
Kitakyushu City, Fukuoka, Japan
mangkhaseum.sackdavong742@mail.kyutech.jp

Yogesh Bhattarai
Department of Civil Engineering
Khwopa College of Engineering
Bhaktapur, Nepal
yogeshbhattarai.sb@gmail.com

Sunil Duwal
Department of Civil Engineering
Khwopa College of Engineering
Bhaktapur, Nepal
duwal.sunil@khwopa.edu.np

Akitoshi Hanazawa
Department of Electrical and Space Systems Engineering
Kyushu Institute of Technology
Kitakyushu City, Fukuoka, Japan
hanazawa@mns.kyutech.ac.jp

***Abstract*— Flood susceptibility mapping is essential for disaster risk management in flood-prone regions. This study utilizes point-based data from publicly available open-source earth system datasets, historical flood observation datasets, Google's cloud computing platform, and deep learning models to create flood susceptibility maps for the Nam Ngum River Basin in Lao PDR. The assessment integrates the digital elevation model (DEM), satellite-observed rainfall data, land use/land cover, and remote sensing image-derived indices such as NDVI to capture detailed flood causation factors. Deep learning techniques, including Artificial neural networks (ANN), Long short-term memory (LSTM), and Deep Neural Networks (DNN), are employed to analyze various hydro-meteorological and geomorphological parameters. The models were trained and tested using eleven flood conditioning variables and 390 locations to ensure predictive accuracy and reliability. These datasets were randomly divided into training and testing datasets in a 70:30 ratio. The developed models are evaluated based on performance metrics like accuracy, precision, and the area under the curve of Receiver Operating Characteristics (AUROC). This map identifies critical zones within the Nam Ngum River Basin at high risk of flooding, offering valuable insights for local authorities and stakeholders. This information is crucial for enhancing flood risk management, emergency planning, and mitigation strategies.**

Keywords—Flood susceptibility modeling, Remote sensing, Deep learning, Google Earth Engine (GEE), Nam Ngum River Basin, Lao PDR

I. Introduction

Flood is a major devastating natural disaster regarding the number of people affected and economic loss [1], [2], [3], [4]. Large and damaging floods are increasingly occurring every year around the world [5]. The frequency and severity of flood events have been exacerbated by climate change, rapid urbanization, and deforestation, necessitating improved flood risk management strategies. Flood susceptibility mapping plays a crucial role in disaster risk management by identifying areas prone to flooding, thereby aiding in the implementation of mitigation measures and emergency preparedness [6].

Recent advancements in remote sensing techniques and deep learning algorithms have significantly enhanced the predictive capabilities of flood susceptibility mapping. These algorithms, such as Artificial Neural Networks (ANNs) [7], [8], Deep Neural Networks (DNNs) [9], [10], and Long Short-Term Memory Networks (LSTMs) [11], [12], [13] can process large datasets, recognize complex patterns, and make highly accurate predictions. Unlike traditional machine learning methods, deep learning models excel at capturing intricate non-linear relationships and temporal dependencies in data, making them particularly suitable for flood susceptibility mapping. These studies are backed by increased computational capabilities, freely available public databases, and open-source platforms such as Google Earth Engine and Python platforms.

This study integrates various freely available geospatial datasets and deep learning techniques, specifically ANN, DNN, and LSTM networks, to create flood susceptibility maps for the Nam Ngum River Basin in Lao PDR. The specific objectives are 1) Utilize Flood Conditioning Factors, a) derived from the ALOS-PALSAR DEM of 12.5 m resolution; these factors include elevation, slope, aspect, curvature, topographic wetness index (TWI), stream power index (SPI), distance to the river (DTR), and b) satellite image-derived normalized difference vegetation index (NDVI), land use/land cover (LULC), remote sensing-based rainfall, and c) soil type data. 2) Train and Validate Deep Learning Models: Train the deep learning models and validate their predictive accuracy and reliability using historical flood data as training and testing datasets from the prepared flood inventory dataset. 3) Generate Flood Susceptibility Maps: Create maps identifying high-risk zones within the basin, providing essential information for local authorities and stakeholders to enhance flood risk management and mitigation strategies.

The main objective of this study is to develop precise flood susceptibility maps using deep learning techniques. These maps are crucial for disaster management, as they help identify areas prone to flooding, enabling better planning, mitigation, and response strategies. In the Nam Ngum River Basin, where floods can have devastating impacts, accurate flood mapping is essential for safeguarding communities and infrastructure.

II. Materials and Methods

A. Description of the Study Area

The Nam Ngum River Basin (NNRB) is a significant tributary of the Mekong River, originating in the northern mountainous region of Xiengkhuang Province, flowing through Vientiane Province, and merging with the Mekong River in

turnitin Page 130 of 204 - Integrity Submission

Submission ID trn:oid:::1:3308505772

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

turnitin Page 131 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Vientiane, the capital city of Laos. The NNRB, the country's fourth-largest river basin, spans 16,800 km^2 between longitudes 102º 25' E and 103º 30' E and latitudes 18º 30' N and 19º 30' N [14], plays a crucial role in the region's socio-economic development by providing water resources for agriculture, hydropower, and domestic use. The total length of the river is approximately 420 km, making up 7.3% of the entire area of Laos and accommodating 9% of the total population in the country. The upper part is hilly and mountainous, while the lower part is flat. From June to October, the rainy season sees heavy rainfall of 1500 to 3000 mm annually, exacerbated by Southwest monsoons and Pacific Ocean typhoons causing floods almost yearly [15], [16].

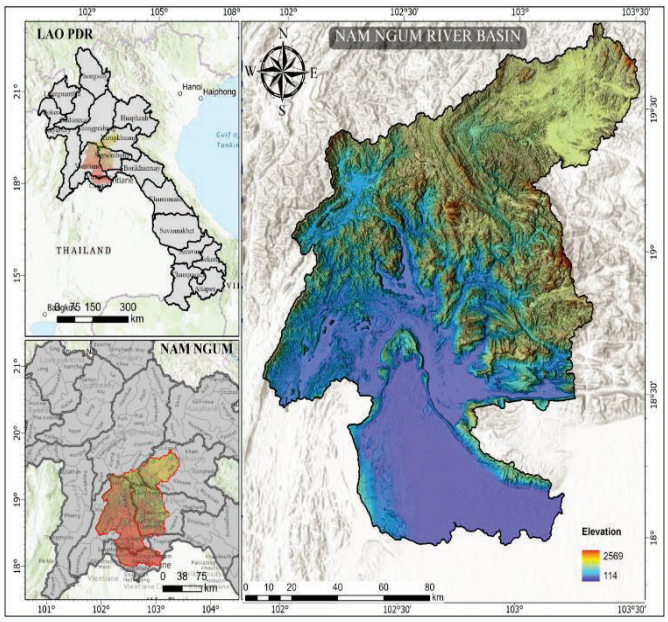


Fig. 1. The location of the Nam Ngum River Basin

B. Flood Inventory Map

The flood inventory map was generated within the GIS environment, utilizing both Google Earth Engine and ArcMap 10.8 based on the Sentinel-1 image (GEE) to enhance flood susceptibility mapping. Sentinel-1 GRD data were accessed and preprocessed in GEE, including radiometric calibration, speckle filtering, and terrain correction. The preprocessing of Sentinel-1 images, accessed through Google Earth Engine (GEE) libraries, encompassed essential procedures such as the application of orbit files, removal of border noise through Analysis Ready Data (ARD), thermal noise mitigation, radiometric correction, terrain correction, and conversion of the backscatter coefficient to decibels. A smoothing filter was applied to the generated image to address the intrinsic speckle effect inherent in radar imagery [17]. The determination of an optimal threshold, set at 1.24, was achieved through iterative experimentation, leading to the segmentation of the raster derived from the change-detection algorithm into distinct categories denoting flooded (1) and non-flooded (0) segments [12], [13]. We selected only 390 past flood points. The non-flood locations were selected visually, where the probability of flooding is none—for example, the hilltops and ridges of the mountains. Equal numbers of flood and non-flood locations were used for the inventory for increased accuracy, as suggested by [8], [18]. To identify flood locations, we used data from the Colorado Flood Observatory and the LAOS knowledge for the development (K4D) portal to verify the flood points located. Values of 1 as flood and 0 as non-flood points were assigned for model training and testing, using 70% of the data for training and 30% for testing.

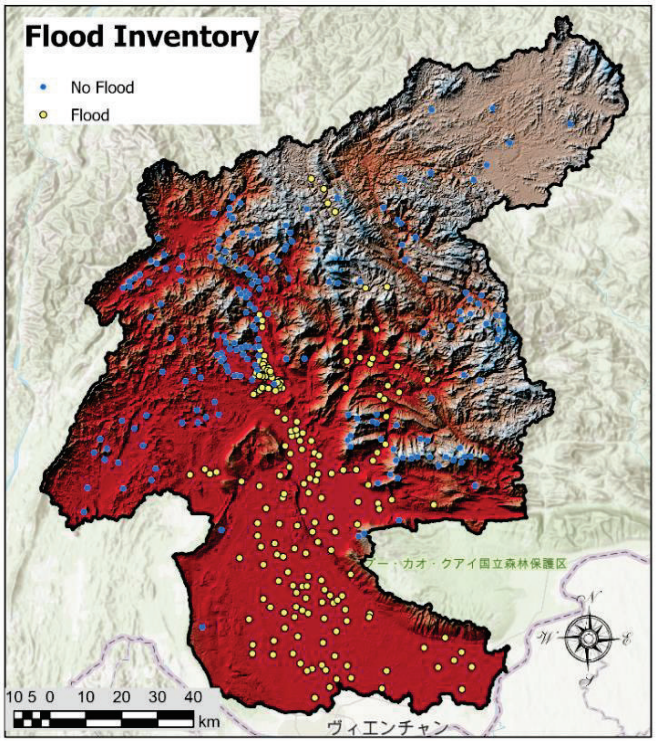


Fig. 2. Flood Inventory Map

C. Flood conditioning factors

We collected geospatial data from various sources, including ALOS-PALSAR DEM for factors like slope and elevation, ERA5 for long-term rainfall, Landsat 8 for NDVI, and Sentinel-2 for land use and cover. The flood inventory map, crucial for flood susceptibility mapping, was generated using these data. Eleven conditioning factors, elevation, slope, curvature, aspect, DTR, drainage density, SPI, and TWI, were selected based on the literature. This study introduces eleven flood conditioning factors:

Elevation, defined as the height difference from sea level, is crucial in flood studies. Higher elevations have a lower flood probability due to runoff moving downstream. The slope is crucial in flooding, as steeper slopes increase water velocity and reduce infiltration, while flat areas are more prone to flooding. Aspect refers to the slope's horizontal direction. Sun-facing areas flood less often, while areas with less sunlight have more soil moisture and runoff, increasing flood risk. Stream Power Index (SPI) indicates a river's erosive power; high SPI means strong flow, while low SPI areas are more prone to flooding.

$$SPI = (\alpha * \tan\beta) \quad [1]$$

where α is the specific basin area, and β is the slope angle in degrees at the point.

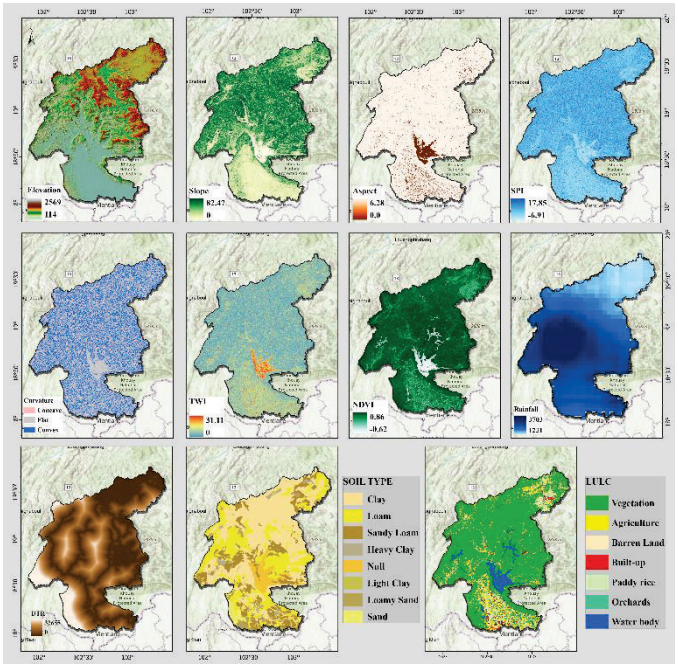


Fig. 3. Flood conditioning factor maps

Curvature indicates ground surface shape: flat, convex, or concave. Flat areas are more prone to flooding. Negative values represent convex, positive values represent concave, and zero represents flat surfaces. Topographic Wetness Index (TWI), developed by [19] measures water accumulation per unit area in a catchment, influenced by gravitational flow. Higher TWI values indicate greater susceptibility to flooding in the area.

$$TWI = \ln\left(\frac{\alpha}{tan\beta}\right) \quad [2]$$

where α is the cumulative area upstream of the point draining to it, and β is the slope angle in degrees. Normalized Difference Vegetation Index (NDVI) is crucial in flood susceptibility studies. Vegetation is a protective barrier against flooding, and NDVI is essential for assessing vegetation cover and density.

$$NDVI = \frac{(NIR-RED)}{(NIR+RED)} \quad [3]$$

where NIR and RED are channel values, NDVI ranges from -1 to +1. Higher values indicate healthier, denser vegetation, which can retain more water. High rainfall increases soil saturation and runoff, raising flood risk, especially in poorly drained or saturated areas. Distance to river is a key factor in flood occurrence; areas near rivers and streams are more susceptible due to their role in flood discharge. Soil type refers to the composition and characteristics of soil that affect its water absorption, retention, and permeability, influencing how water interacts with the ground during rainfall and flooding and thus determining the severity and extent of floods in a region. Land use and land cover (LULC) influences flooding through subsidence, runoff, and evapotranspiration. , this factor was extracted from Landsat-8 OLI into seven classes. Vegetated areas promote water infiltration, mitigating runoff, while built-up areas intensify it.

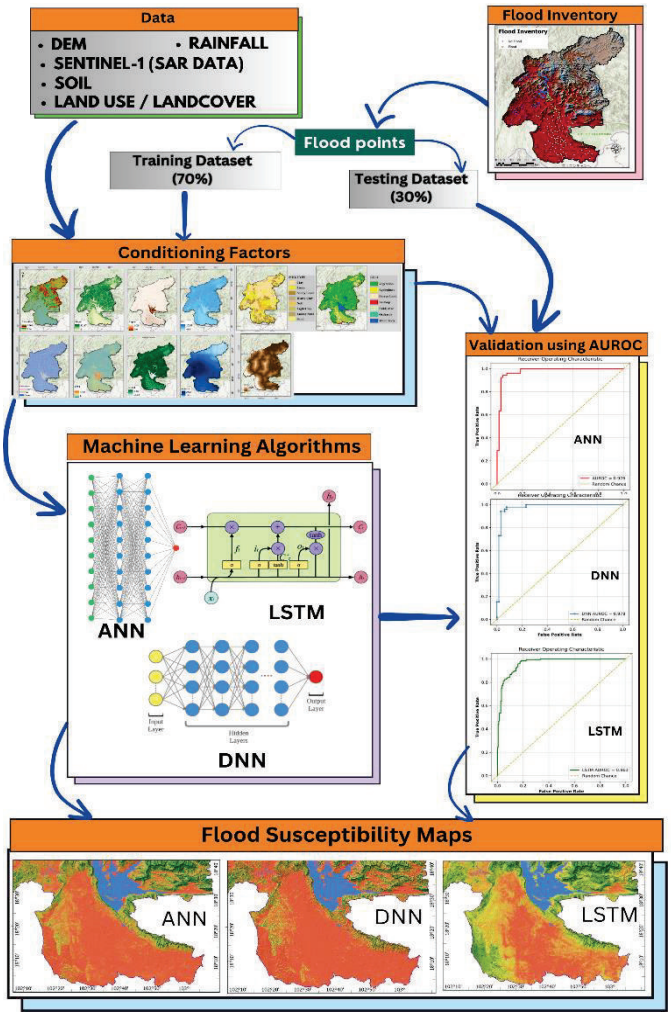


Fig. 4. Detailed methodology

D. Deep learning methods

1) Artificial Neural Network

ANN models replicate the brain's interconnected neurons, processing sensory inputs through layers of artificial nodes. Each connection between layers carries weighted information, influencing the final output. Researchers value ANNs for their nonlinear modeling abilities and adaptability to complex systems [7], [8].

2) Long Short-Term Memory

LSTM is a type of recurrent neural network (RNN) designed to handle the issue of vanishing gradients, enabling it to effectively capture long-term dependencies in sequential data by selectively remembering or forgetting information over time [11], [12], [13].

3) Deep Neural Network

DNN is an artificial neural network with multiple layers between the input and output layers. These networks consist of an input layer, several hidden layers, and an output layer. Each layer comprises nodes (neurons) that process and transform the input data. The 'deep' aspect refers to the network's depth, i.e., the number of hidden layers, which enables it to model complex patterns and relationships in data [9], [10].

turnitin Page 133 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

E. Performance evaluation of models

The models were optimized using hyperparameter tuning to find the best parameter values. In flood susceptibility mapping, correctly identified flood and non-flood pixels are True Positives (TP) and True Negatives (TN), while misclassifications are False Positives (FP) and False Negatives (FN). AUROC assesses model performance by measuring separability [13], using sensitivity (true positive rate) on the y-axis and 1 − specificity (false positive rate) on the x-axis. A higher AUROC value closer to 1 indicates better model performance, while 0.5 suggests poor accuracy [8], [12], [13]. Evaluation metrics like AUROC (Equation 7), Kappa Score, and Accuracy were used to assess the model's parameters.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FM} \quad [4]$$

$$Sensitivity = \frac{TP}{TP+FN} \quad [5]$$

$$Specificity = \frac{TP}{FP+TN} \quad [6]$$

$$AUROC = \sum TP + \frac{\sum TN}{P} + N \quad [7]$$

F. Flood Susceptibility mapping

Flood susceptibility mapping is one of the significant steps in identifying flood-prone areas. Using publicly available open-source data and methodologies such as deep learning has been a game changer in flood studies. Therefore, ANN, DNN, and LSTM models were trained and tested using scikit-learn, TensorFlow, and Keras platforms to simulate flood susceptibility in Nam Ngum River Basin, Lao PDR. The trained model is applied to stacked flood conditioning factors raster to generate flood susceptibility maps [18], [19]. The generated maps are classified using the natural breaks method into five flood susceptibility classes (Very high, high, medium, low, and very low).

III. Result and Discussion

TABLE I. The optimum values of the tuning hyperparameters of ANN, DNN, and LSTM

Method	Hyperparameter	Optimum Value
ANN	'batch_size' 'epochs' 'Optimizer_Trial' 'Neurons_Trial'	8, 32, 64 5,10 'adam', 'rmsprop' 10,15
DNN	'batch_size' 'epochs' 'Optimizer_Trial' 'Neurons_Trial'	64 100, 500 'adam' 64
LSTM	'batch_size' 'epochs' 'Optimizer_Trial' 'Neurons_Trial'	10, 32, 64 10, 20 'rmsprop' 40

The hyperparameters, as shown in Table I, were used for the flood susceptibility mapping in this study. The hyperparameter optimization reflects the distinct nature and requirements of each model. ANN benefits from quicker and smaller-scale training setups, DNN from more extended training periods and complexity, and LSTM from a balanced approach catering to temporal dependencies.

Figure 5 summarizes the performance parameter values of the models, including accuracy, precision, F1-score, and Kappa score, offering a thorough evaluation of their predictive abilities for flood susceptibility mapping. AUROC values for the DNN model stood out, with the highest value of 0.978, showing its excellent performance in predicting flood-prone areas. The ANN and LSTM models followed with AUROC of 0.976 and 0.963 (Figure 5 and Figure 6). However, ANN shows the best overall performance with the highest precision, recall, accuracy, F1-score, and Kappa. It also has a slightly lower AUROC than DNN but is better. DNN performs very closely to ANN with slight variations but has the highest AUROC, indicating its excellent ability to distinguish between classes. LSTM has the lowest performance among the three models, with lower true positives and recall, which affects its overall accuracy and Kappa. However, it still maintains a good AUROC. In summary, ANN is the most balanced and high-performing model, followed closely by DNN. LSTM, while still effective, falls slightly behind in several key performance metrics. The comparison of the result with the flood database from the Colorado Flood Observatory, the Knowledge for Development (K4D), and historical flood observations using Sentinel-1 SAR image showed that the results are appropriate for ANN and DNN models.

Parameter	ANN	DNN	LSTM
Precision	0.95	0.94	0.93
Recall	0.94	0.94	0.88
Accuracy	0.94	0.94	0.93
F1-Score	0.95	0.94	0.94
Kappa	0.879	0.862	0.861
AUROC	0.976	0.978	0.963

Fig. 5. Performance of the parameter values of the model

The results of the three deep learning models are presented in Figure 8. The study revealed that for the ANN model, 27% (4582 km^2) of the area lies in a very high flood susceptibility zone, followed by 9% (1634 km^2) in highly susceptible, 8% (1338 km^2) in medium, 12% (1981 km^2) in less and 44% (7397 km^2) in very less flood susceptible zones. Similarly, DNN shows that 29% (4832 km^2) of the area lies under very high susceptible areas, 19% (3189) in high susceptible areas, 11% (1919 km^2) in medium, 14% (2395 km^2) in less, and 27%

turnitin Page 134 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

(4596 km^2) in very less flood susceptible areas. Lastly, LSTM shows that 38% (6368 km^2) lies in very high, 15% (2561 km^2) in high, 13% (2219 km^2) in medium, 14% (2399 km^2) in less, and 20% (3385 km^2) in very less susceptible areas. It can be observed that the variations in parameter values and flood susceptible areas are less for ANN and DNN compared to LSTM. In this scenario, it can be concluded that the results of ANN and DNN are more reliable than those of LSTM. However, the result from LSTM should also be considered for flood susceptibility mapping since the parameters are in the higher range. In this scenario, it should be noted that 36% to 56% of the area of NNRB is highly susceptible to flood. This alarming condition depicts a need for serious concern in flood disaster prevention and management strategies.

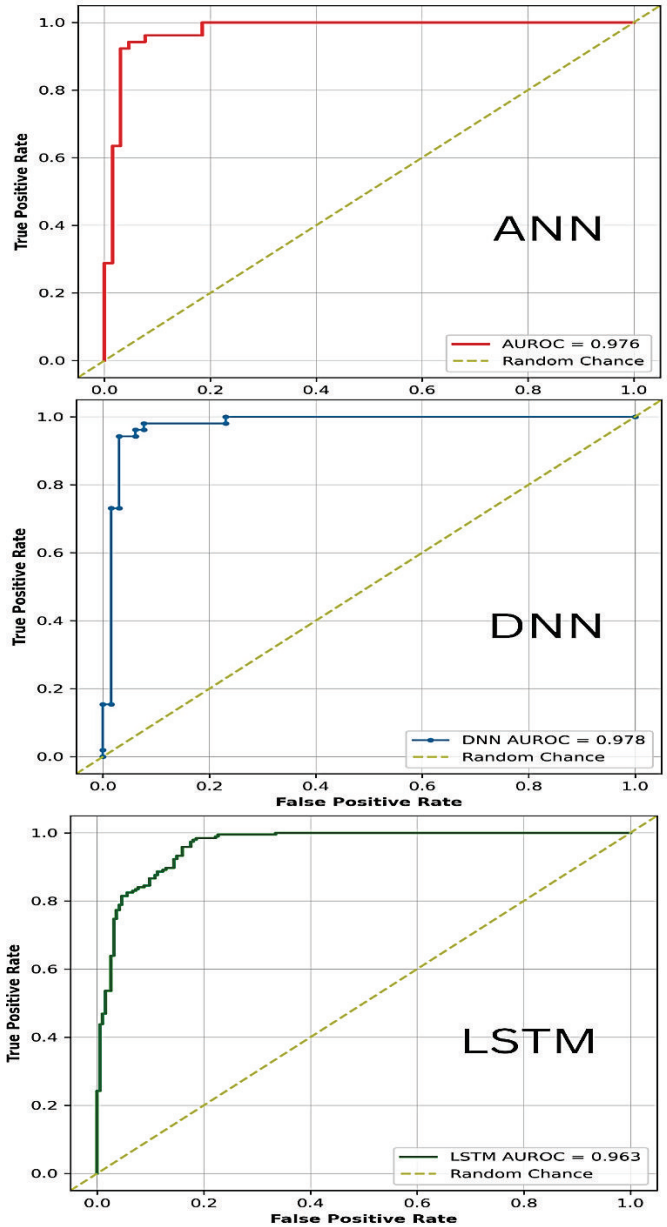


Fig. 6. Area Under the Curve for Receiver Operating Characteristics (AUROC)

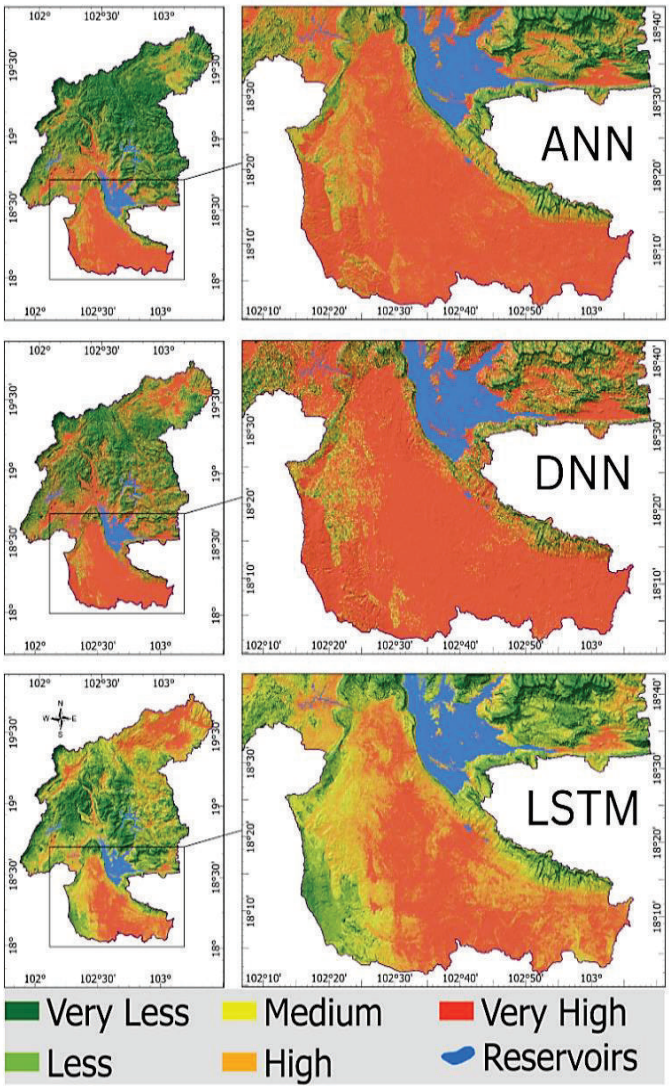


Fig. 7. Flood Susceptibility Map

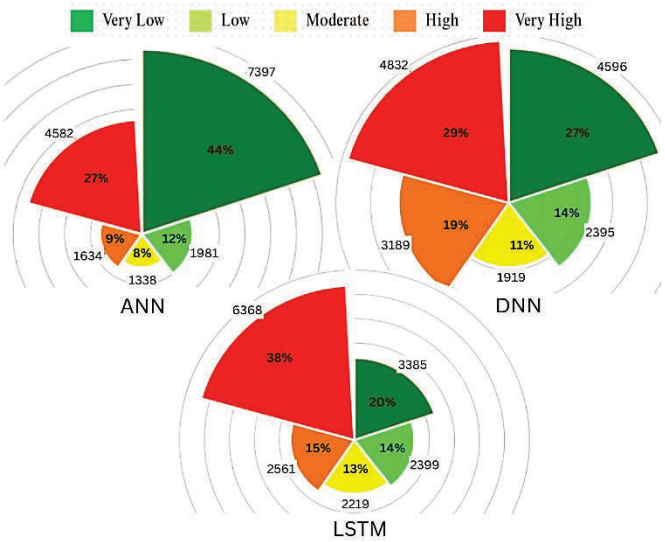


Fig. 8. Flood susceptible area in NNRB predicted by different deep learning algorithms

turnitin Page 134 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

turnitin Page 135 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Naturally, a flood occurs in areas of relatively low elevation and slope, as in the previous studies [8], [20], [21]. According to Figure 7, The NNRB has a flat region characterized by low elevation and slopes in the downstream southern part of the basin before joining the Mekong River. It is relatively narrow; the bed slope is very mild, and the influence of the Mekong River level causes difficulty draining the flooded area. When floods occur, low-lying regions can serve as natural basins, accumulating and holding water, which prolongs the flooding and amplifies the potential damage [13], [22]. In addition, there are also many low-lying areas in the Vientiane Plain, which are inundated by small floods. Even though many hydropower dams were constructed upstream of the flat plain, the downstream area is flooded almost yearly [23], [24].

IV. Conclusion

In our study focusing on enhancing the precision and efficiency of flood modeling in Nam Ngum River Basin, Lao PDR, we proved the power of using Sentinel 1 SAR imagery in the GEE platform to accurately identify and map flood locations in the past. The evaluation of models used three deep learning models (ANN, DNN, and LSTM) to predict flood susceptibility in NNRB, an area annually affected by typhoons and heavy rainfall. The models were trained and tested using eleven flood conditioning variables and 390 locations. The DNN and ANN performed better than LSTM, as evidenced by their AUROC precision, F1-score, accuracy, and kappa values. The study's results offer valuable insights for flood risk assessment and developing effective flood control plans. As floods threaten infrastructure, agriculture, and the economy worldwide, these deep learning insights using open-source big data platforms with the Google Earth Engine database can aid local authorities, planners, policymakers, and stakeholders in disaster risk management and climate change mitigation. Future research works can further explore a hybrid combining hydrodynamic deep learning-based modeling to predict flood susceptible area, velocity, and depth of flooding, enabling more comprehensive modeling of flood vulnerability in Nam Ngum River Basin (NNRB), Lao PDR.

Reference

[1] G. Di Baldassarre, A. Montanari, H. Lins, D. Koutsoyiannis, L. Brandimarte, and G. Blöschl, "Flood fatalities in Africa: From diagnosis to mitigation," *Geophys. Res. Lett.*, vol. 37, no. 22, 2010, doi: 10.1029/2010GL045467.

[2] G. FitzGerald, W. Du, A. Jamal, M. Clark, and X.-Y. Hou, "Flood fatalities in contemporary Australia (1997–2008)," *Emerg. Med. Australas.*, vol. 22, no. 2, pp. 180–186, 2010, doi: 10.1111/j.1742-6723.2010.01284.x.

[3] E. N. Rappaport, "Fatalities in the United States from Atlantic Tropical Cyclones: New Data and Interpretation," *Bull. Am. Meteorol. Soc.*, vol. 95, no. 3, pp. 341–346, Mar. 2014, doi: 10.1175/BAMS-D-12-00074.1.

[4] U. Khalil and N. M. Khan, "Floodplain Mapping for Indus River: Chashma –Taunsa Reach," 2017.

[5] A. Morrison, C. j. Westbrook, and B. f. Noble, "A review of the flood risk management governance and resilience literature," *J. Flood Risk Manag.*, vol. 11, no. 3, pp. 291–304, 2018, doi: 10.1111/jfr3.12315.

[6] S. A. Shah and S. Ai, "Flood susceptibility mapping contributes to disaster risk reduction: A case study in Sindh, Pakistan," *Int. J. Disaster Risk Reduct.*, vol. 108, p. 104503, Jun. 2024, doi: 10.1016/j.ijdrr.2024.104503.

[7] S. Priscillia, C. Schillaci, and A. Lipani, "Flood susceptibility assessment using artificial neural networks in Indonesia," *Artif. Intell. Geosci.*, vol. 2, pp. 215–222, Dec. 2021, doi: 10.1016/j.aiig.2022.03.002.

[8] S. Duwal, D. Liu, and P. M. Pradhan, "Flood susceptibility modeling of the Karnali river basin of Nepal using different machine learning approaches," *Geomat. Nat. Hazards Risk*, vol. 14, no. 1, Dec. 2023, doi: 10.1080/19475705.2023.2217321.

[9] R. Costache, P. T. T. Ngo, and D. T. Bui, "Novel Ensembles of Deep Learning Neural Network and Statistical Learning for Flash-Flood Susceptibility Mapping," *Water*, vol. 12, no. 6, p. 1549, May 2020, doi: 10.3390/w12061549.

[10] D. Tien Bui *et al.*, "A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area," *Sci. Total Environ.*, vol. 701, p. 134413, Jan. 2020, doi: 10.1016/j.scitotenv.2019.134413.

[11] Z. Fang, Y. Wang, L. Peng, and H. Hong, "Predicting flood susceptibility using LSTM neural networks," *J. Hydrol.*, vol. 594, pp. 125734–125734, 2021, doi: 10.1016/j.jhydrol.2020.125734.

[12] Y. Bhattarai, S. Duwal, S. Sharma, and R. Talchabhadel, "Leveraging machine learning and open-source spatial datasets to enhance flood susceptibility mapping in transboundary river basin," *Int. J. Digit. Earth*, vol. 17, no. 1, p. 2313857, Dec. 2024, doi: 10.1080/17538947.2024.2313857.

[13] S. Mangkhaseum, Y. Bhattarai, S. Duwal, and A. Hanazawa, "Flood susceptibility mapping leveraging open-source remote-sensing data and machine learning approaches in Nam Ngum River Basin (NNRB), Lao PDR," *Geomat. Nat. Hazards Risk*, vol. 15, no. 1, p. 2357650, Dec. 2024, doi: 10.1080/19475705.2024.2357650.

[14] T. Meema, Y. Tachikawa, Y. Ichikawa, and K. Yorozu, "Uncertainty assessment of water resources and long-term hydropower generation using a large ensemble of future climate projections for the Nam Ngum River in the Mekong Basin," *J. Hydrol. Reg. Stud.*, vol. 36, p. 100856, Aug. 2021, doi: 10.1016/j.ejrh.2021.100856.

[15] S. Dhungana, S. Shrestha, T. P. Van, S. Kc, A. Das Gupta, and T. P. L. Nguyen, "Evaluation of gridded precipitation products in the selected sub-basins of Lower Mekong River Basin," *Theor. Appl. Climatol.*, vol. 151, no. 1–2, pp. 293–310, Jan. 2023, doi: 10.1007/s00704-022-04268-1.

[16] R. Bartlett, J. Baker, G. Lacombe, S. Douangsavanh, and M. Jeuland, "Analyzing Economic Tradeoffs of Water Use in the Nam Ngum River Basin, Lao PDR," *SSRN Electron. J.*, 2012, doi: 10.2139/ssrn.2469222.

[17] S. Mehravar, S. V. Razavi-Termeh, A. Moghimi, B. Ranjgar, F. Foroughnia, and M. Amani, "Flood susceptibility mapping using multi-temporal SAR imagery and novel integration of nature-inspired algorithms into support vector regression," *J. Hydrol.*, vol. 617, p. 129100, Feb. 2023, doi: 10.1016/j.jhydrol.2023.129100.

[18] A. R. M. Towfiqul Islam *et al.*, "Flood susceptibility modelling using advanced ensemble machine learning models," *Geosci. Front.*, vol. 12, no. 3, p. 101075, May 2021, doi: 10.1016/j.gsf.2020.09.006.

[19] K. J. BEVEN and M. J. KIRKBY, "A physically based, variable contributing area model of basin hydrology / Un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant," *Hydrol. Sci. Bull.*, vol. 24, no. 1, pp. 43–69, Mar. 1979, doi: 10.1080/02626667909491834.

[20] K. Khosravi, H. R. Pourghasemi, K. Chapi, and M. Bahri, "Flash flood susceptibility analysis and its mapping using different bivariate models in Iran: a comparison between Shannon's entropy, statistical index, and weighting factor models," *Environ. Monit. Assess.*, vol. 188, no. 12, p. 656, Nov. 2016, doi: 10.1007/s10661-016-5665-9.

[21] S. Talukdar *et al.*, "Flood susceptibility modeling in Teesta River basin, Bangladesh using novel ensembles of bagging algorithms," *Stoch. Environ. Res. Risk Assess.*, vol. 34, no. 12, pp. 2277–2300, 2020, doi: 10.1007/s00477-020-01862-5.

[22] K. M. Al-Kindi and Z. Alabri, "Investigating the Role of the Key Conditioning Factors in Flood Susceptibility Mapping Through Machine Learning Approaches," *Earth Syst. Environ.*, Jan. 2024, doi: 10.1007/s41748-023-00369-7.

[23] V. Keophila, "Multi-objective optimization for flood control operation and electricity production of Nam Ngum 1 and 2 hydropower plants," *J. Thai Interdiscip. Res.*, vol. 13, p. 58, 2018, doi: 10.14456/JTIR.2018.52.

[24] V. Keophila, A. Promwungkwa, and K. Ngamsanroaj, "Effectiveness of Cascades Reservoir for Flood Control Operation and Electricity Production in Nam Ngum River," *J. Phys. Conf. Ser.*, vol. 1175, p. 012276, Mar. 2019, doi: 10.1088/1742-6596/1175/1/012276.

turnitin Page 136 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Enhancing Short Text Semantic Similarity Measurement Using Pretrained Word Embeddings and Big Data

Supakpong Jinarat
College of Innovative Technology and Engineering
Dhurakij Pundit University
Bangkok Thailand.
supakpong.jin@dpu.ac.th

Ratchakoon Pruengkarn
College of Innovative Technology and Engineering
Dhurakij Pundit University
Bangkok Thailand.
ratchakoon.prn@dpu.ac.th

***Abstract*— Measuring semantic similarity between short texts is a fundamental task in natural language processing with applications in information retrieval, question answering, and machine translation. Traditional methods such as term frequency (tf) and term frequency-inverse document frequency (tfidf) rely on lexical matching and fail to capture semantic meanings. This paper introduces Word Embeddings for Semantic Similarity (WESS), leveraging pretrained Word2Vec embeddings to capture semantic relationships. We fine-tune the Word2Vec model on the Quora Question Pairs dataset, containing 404,290 pairs of questions labeled as duplicate or non-duplicate. Our approach calculates text similarity by aggregating word embedding similarities. Experimental results demonstrate that WESS outperforms traditional methods, achieving an accuracy of 0.675, a 9.4% improvement over tf (0.617), a 6.5% improvement over tfidf (0.634), a 3.8% improvement over tfidf combined with Word2Vec (0.636), and a 1.9% improvement over standalone Word2Vec (0.662). These findings underscore the importance of semantic understanding in text similarity tasks and validate the effectiveness of pretrained word embeddings for capturing nuanced semantic relationships. The SSST method offers a robust and accurate approach for measuring semantic similarity between short texts, providing significant improvements over traditional approaches.**

Keywords—short text, semantic similarity, word embeddings

I. INTRODUCTION

Measuring semantic similarity between short texts is a fundamental task in natural language processing (NLP) with wide-ranging applications in information retrieval, question answering, machine translation, and more. Traditional methods for text similarity have primarily relied on lexical matching techniques, such as term frequency-inverse document frequency (TF-IDF), Jaccard similarity, and cosine similarity, which compare texts based on the presence or absence of words.

TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a corpus. It combines the term frequency (the number of times a term appears in a document) and the inverse document frequency (which measures how common or rare a term is across the entire corpus). The cosine similarity metric is often applied to TF-IDF vectors to compute the similarity between documents by measuring the cosine of the angle between two vectors in a multi-dimensional space [8].

Jaccard similarity measures the similarity between two sets by dividing the size of the intersection by the size of the union of the sets. For text similarity, the sets are usually composed of the unique words in each text. While simple, this method only captures lexical overlap and ignores the semantic meaning of words [9].

However, these traditional methods have significant limitations. They are sensitive to variations in word choice and often fail to capture the underlying semantic meaning of texts. For instance, synonyms or paraphrased content might be deemed dissimilar due to differing lexical compositions. This is particularly problematic for short texts, where limited context further exacerbates the inadequacies of lexical-based methods.

To overcome these challenges, more sophisticated approaches have been developed. Latent semantic analysis (LSA) is one such method that attempts to capture semantic information by transforming text data into a lower-dimensional space, revealing hidden relationships between words and documents [3]. Similarly, latent Dirichlet allocation (LDA) models texts as mixtures of topics, offering another way to infer semantic content from text [1].

Despite these advances, the rise of deep learning and the availability of large-scale datasets have paved the way for even more powerful methods of semantic similarity measurement. Word embeddings, such as Word2Vec [5], GloVe [6], and FastText [2], represent words in continuous vector spaces where semantically similar words are mapped to nearby points. These models are trained on vast corpora, enabling them to capture rich semantic relationships between words.

Building on word embeddings, recent advancements have introduced contextualized word embeddings like ELMo [7], BERT [4] [9], and its variants, which further enhance semantic understanding by considering the context in which words appear. These models leverage large-scale pre-training on diverse text corpora followed by fine-tuning on specific tasks, setting new benchmarks in various NLP applications.

This paper explores the application of word embeddings models to measure semantic similarity between short texts, leveraging big data to train and fine-tune these models for enhanced performance. By integrating advanced word embeddings with robust semantic similarity algorithms, we aim to address the limitations of traditional methods and improve the accuracy of short text similarity measurement.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

II. RELATED WORK

The task of measuring semantic similarity between short texts has been extensively studied in the field of natural language processing (NLP). Traditional methods have primarily relied on lexical matching techniques, which compare texts based on the presence or absence of words. However, these methods often fail to capture the deeper semantic meanings of the texts, prompting the development of more sophisticated approaches.

A. Traditional Methods

Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity are among the earliest techniques used for text similarity. [8] introduced these methods, which represent documents as vectors in a high-dimensional space where each dimension corresponds to a term's importance in the document relative to a corpus. While effective for certain tasks, these approaches are limited in their ability to handle semantic relationships between words.

Jaccard similarity is another traditional method that measures the similarity between two sets by dividing the size of their intersection by the size of their union. This method has been widely used for comparing short texts but suffers from similar limitations as TF-IDF, as it does not account for the semantic content of the words [9].

B. Latent Semantic Analysis and Topic Models

To address the shortcomings of lexical matching, Latent Semantic Analysis (LSA) was developed, which reduces the dimensionality of text data and uncovers latent structures in the relationships between words and documents. [3] demonstrated that LSA could capture semantic meanings by mapping words and documents into a lower-dimensional space.

Building on this idea, Latent Dirichlet Allocation (LDA) was proposed by [1] as a generative probabilistic model for discovering topics in a collection of documents. LDA models each document as a mixture of topics, allowing it to infer the semantic content more effectively than simple word matching techniques.

C. Word Embeddings

The introduction of Word2Vec by [5] marked a significant advancement in the field of word embeddings. Word2Vec uses neural networks to learn vector representations of words from large text corpora, positioning semantically similar words close to each other in the vector space. This model, with its CBOW and Skip-gram architectures, has been pivotal in capturing word meanings and relationships.

Following Word2Vec, GloVe (Global Vectors for Word Representation) was developed by [6]. GloVe combines the advantages of global matrix factorization and local context window methods, producing word vectors that capture both semantic and syntactic properties [6]. Similarly, FastText, introduced by [2], enhances word embeddings by incorporating subword information, thus improving the representation of rare and out-of-vocabulary words.

III. METHOD

In this section, we detail our proposed method for measuring short text semantic similarity using word embeddings. Our approach, Word Embeddings for Sematic Similarity (WESS), leverages pretrained word embedding models to capture the semantic meanings of words and then calculates the similarity between two texts based on the aggregated similarity of their word embeddings.

A. Word Embeddings Model

Word embeddings are dense vector representations of words that capture semantic meaning by positioning semantically similar words close to each other in the vector space. We use pretrained word embedding models, such as Word2Vec, which are trained on large corpora and have demonstrated strong performance in capturing semantic relationships.

For our experiments, we employ a pretrained Word2Vec model, which is further fine-tuned on the Quora Question Pairs dataset to adapt the embeddings to the specific domain of question similarity. This fine-tuning process helps to capture the contextual nuances specific to the dataset, improving the accuracy of the similarity measurements.

B. Text Representation

To represent each text, we follow these steps:

1) Tokenization: Each text is tokenized into individual words.

2) Embedding Lookup: For each word in the text, we retrieve its corresponding word embedding from the pretrained Word2Vec model.

3) Handling Out-of-Vocabulary Words: Words that are not present in the pretrained model's vocabulary are handled by using a special embedding vector (e.g., a zero vector) or by ignoring them.

C. Calculating Text Similarity

Once we have the word embeddings for both texts, we calculate the similarity between the two texts using the following approach:

1) Word Pair Similarity: For each word in the first text, we find the most similar word in the second text based on cosine similarity between their embeddings. The cosine similarity between two vectors $v1$ and $v2$ is given by:

$$\text{cosine_similarity}(\mathbf{v_1}, \mathbf{v_2}) = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|} \quad (1)$$

2) Aggregate Similarity: The overall similarity between the two texts is calculated by averaging the maximum word pair similarities. This approach ensures that each word in the first text is matched with its most semantically similar counterpart in the second text, and the similarities are aggregated to provide a final similarity score.

Let ***T1*** and ***T2*** be the sets of word embeddings for the two texts. $\boldsymbol{w_i}$ is a word of the text ***Ti***. The similarity score S is calculated as follows:

$$S(\text{T1}, \text{T2}) = \frac{1}{|\text{T1}|} \sum_{\mathbf{w_1} \in \text{T1}} \max_{\mathbf{w_2} \in \text{T2}} \text{cosine_similarity}(\mathbf{w_1}, \mathbf{w_2}) \quad (2)$$

3) Symmetry: To ensure symmetry in similarity measurement, the same process is repeated with the roles of

the texts reversed, and the final similarity score is obtained by averaging the two scores.

$$S_{\text{final}}(\text{T1}, \text{T2}) = \frac{S(\text{T1}, \text{T2}) + S(\text{T2}, \text{T1})}{2} \quad (3)$$

IV. Experiments

A. Dataset

For our experiments, we utilized the Quora Question Pairs dataset [10], a widely-used benchmark for evaluating the performance of text similarity models. The dataset contains 404,290 question pairs, each labeled as either duplicate or non-duplicate, indicating whether the questions in the pair have the same meaning. The questions cover a wide range of topics and exhibit considerable variability in length, syntactic structure, and vocabulary, providing a challenging benchmark for text similarity models. Approximately 37% of the pairs are labeled as duplicates, reflecting a slight imbalance in the dataset.

B. Data Preprocessing

Prior to training our models, we performed several preprocessing steps to ensure the data was in an optimal format for our experiments:

1) Tokenization: Each question was tokenized into individual words using the NLTK tokenizer.

2) Lowercasing: All text was converted to lowercase to maintain consistency and reduce variability caused by case differences.

3) Removal of Special Characters: Special characters and punctuation were removed to focus on the textual content.

4) Padding and Truncation: Sentences were padded or truncated to a fixed length to create uniform input sizes for the models.

C. Experimental Setup

We evaluated the performance of various methods for short text similarity measurement on the Quora Question Pairs dataset. The methods compared include:

- Term Frequency (tf)
- Term Frequency-Inverse Document Frequency (tfidf)
- Word2Vec (word2vec)
- Combination of tfidf and Word2Vec (tfidf_word2vec)
- Semantic Word2Vec (semantic_word2vec)

Each method was implemented and trained using the preprocessed Quora Question Pairs dataset. The accuracy of each method was measured by comparing the predicted similarity labels with the ground truth labels provided in the dataset.

For the implementation of Word2Vec, we used the Gensim [1] library to train the embeddings on the Quora Question Pairs dataset. The embeddings were then averaged to create a vector representation for each question. For the combination of tfidf and Word2Vec, we first computed tfidf vectors for the questions and then concatenated these vectors with the Word2Vec embeddings.

For the semantic Word2Vec model, we utilized a pre-trained Word2Vec model fine-tuned on a large corpus of text to capture more nuanced semantic relationships. The fine-tuning process involved further training on the Google News Corpus to adapt the embeddings to the specific domain.

D. Evaluation Metrics

The primary evaluation metric used to compare the performance of the different methods was accuracy. Accuracy was calculated as the ratio of correctly predicted labels to the total number of labels. This metric provides a straightforward measure of how well each model can distinguish between semantically equivalent and non-equivalent question pairs.

V. Results and Discussion

In this section, we present the experimental results of various methods for measuring short text semantic similarity. The methods evaluated include term frequency (tf), term frequency-inverse document frequency (tfidf), Word2Vec, a combination of tfidf and Word2Vec, and semantic Word2Vec.

A. Results

The performance of each method is measured in terms of accuracy. The results are summarized in Table 1.

TABLE I. Accuracy of different methods for short text semantic similarity measurement.

Method	Accuracy
tf	0.617
tfidf	0.634
word2vec	0.662
tfidf_word2vec	0.636
WESS	0.675

B. Discussion

From the experimental results, we observe the following trends:

Term Frequency (tf): The basic tf method achieves an accuracy of 0.617. This method relies solely on the frequency of terms in the text, and does not capture the semantic meaning of the words.

Term Frequency-Inverse Document Frequency (tfidf): The tfidf method shows an improvement over tf, with an accuracy of 0.634. By considering the importance of terms across the corpus, tfidf provides a more refined representation of the text.

Word2Vec: Using Word2Vec embeddings significantly improves the performance, achieving an accuracy of 0.662. Word2Vec captures the semantic relationships between words, resulting in better similarity measurements.

Combination of tfidf and Word2Vec (tfidf_word2vec) yields an accuracy of 0.636. Although this combination

[1] Python Gensim Library: https://radimrehurek.com/gensim/

slightly improves upon tfidf alone, it does not outperform the standalone Word2Vec model. This suggests that combining tfidf with Word2Vec may introduce some redundancy or noise.

Word Embeddings for Semantic Similarity (wess): The semantic Word2Vec model achieves the highest accuracy of 0.675. By leveraging semantic information, this method captures the meanings of words in context, resulting in the most accurate similarity measurements among the evaluated methods.

VI. Conclusion

The experimental results demonstrate the superiority of word embeddings, particularly Word2Vec and its semantic variant, for measuring short text semantic similarity. Traditional methods like tf and tfidf, while useful, fall short in capturing the nuanced meanings of words. The semantic Word2Vec model, which incorporates contextual information, provides the best performance, highlighting the importance of semantic understanding in text similarity tasks.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993-1022, 2003.

[2] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, vol. 41, no. 6, pp. 391-407, 1990.

[4] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proceedings of NAACL-HLT, 2019.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proceedings of the International Conference on Learning Representations (ICLR), 2013.

[6] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[7] M. E. Peters et al., "Deep Contextualized Word Representations," in Proceedings of NAACL-HLT, 2018.

[8] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, New York, NY, USA: McGraw-Hill, 1983.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019.

[10] Quora. "Quora Question Pairs." [Online]. Available: https://www.kaggle.com/c/quora-question-pairs. [Accessed: June. 24, 2024].

turnitin Page 140 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Enhancing Durian Cultivation Efficiency through Data-Driven Smart Farming Using Cluster Analysis and Machine Learning

1st Pattharaporn Thongnim
Department of Mathematics
Faculty of Science
Burapha University
Chonburi, Thailand
pattharaporn@buu.ac.th

2nd Jakkrapan Sreekajon
Information Technology and Data Science
Faculty of Science and Arts
Burapha University
Chanthaburi, Thailand
jsreekajon@gmail.com

3rd Thanaphon Pukseng*
Data Center
Faculty of Science and Arts
Burapha University
Chanthaburi, Thailand
thanaph@buu.ac.th

***Abstract*—This study explores the application of k-means clustering, combined with the elbow and silhouette methods, to analyze durian farm yields and production areas in Eastern Thailand from 2012 to 2023. The aim is to identify optimal farming practices and land use patterns to enhance productivity and sustainability. Using data on yield and area of production, the analysis reveals distinct clusters representing different farming characteristics. The evaluation metrics, including the Davies-Bouldin Index and Dunn Index, indicate that the Elbow method generally provides better defined clusters, although the Silhouette method occasionally shows superior clustering quality. The results show significant shifts in cluster centroids over the years, reflecting changes in smart farming. These insights suggest targeted interventions to optimize resource allocation and improve farm management.**

***Index Terms*—k-means, clustering, durian, elbow method, silhouette score**

I. Introduction

Agriculture is very important in Southeast Asia, especially in Thailand. Eastern Thailand, in particular, benefits from a favorable climate [1] and fertile soil that are ideal for growing a wide variety of crops, including rice, rubber, and fruits [2]. This region significantly contributes to the country's agricultural exports, making it a key player in the national economy. However, increasing the productivity and quality of these farms is a challenging task. Farmers must navigate complex environmental conditions such as varying rainfall patterns [3] and soil health [4], and employ effective farming methods. This requires a deep understanding of the local ecosystem as well as the adoption of advanced agricultural techniques and technologies to enhance yield and ensure sustainable practices.

Durian, known as the King of Fruits is very important in Thailand [5], [6], especially in Chanthaburi. In Thailand, durian cultivation covers extensive areas particularly in Eastern Thailand, which has the right climate and water conditions for this crop. The region conditions help produce a lot of crops and high yields. Durian farming is very important for the agricultural countries and helps the national economy. However, it is hard to make durian farms more productive [7]. It requires the good understanding of the environmental conditions and effective farming methods to optimize agricultural yield and production.

Agricultural data can be used with supervised and unsupervised learning models to group similar parts of farms based on factors such as soil quality, water availability, technology and production levels [8], [9]. This helps farmers understand and manage their farms better. Moreover, precision farming techniques can be used. Farmers can give specific nutrients and set irrigation schedules based on the needs of different parts of the farm [10]. Moreover, this approach makes farming more efficient and sustainable using these data driven techniques can change durian farming and making it more productive.

K-means clustering algorithms are effective at uncovering hidden patterns in agricultural data [11]. These methods group data based on similar characteristics. It can identify farm areas that require specific farming practices. For instance, k-means can divide a farm into zones with similar production needs enhancing resource efficiency [12], [13]. To determine the optimal number of clusters techniques can be used such as the elbow method and the silhouette score [14]. The elbow method involves making a plot of the sum of squared distances from each point to its cluster center. An elbow point is found where the decrease rate slows down sharply. Therefore, this point shows the number of clusters. In addition, the silhouette score checks how similar an object is to its own cluster compared to other clusters. It can be seen that higher scores is better for clustering.

This study aims to answer the following research questions. How can k-means clustering along with the elbow and silhouette methods are used to effectively analyze durian farm yields and production areas in eastern Thailand? What is the best agricultural practices that can be found through this clustering method? How can the insights from this data analysis help make durian farming more sustainable and boost economic growth in Thailand?

Therefore, this paper explores using k-means clustering with

*corresponding author

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

turnitin Page 141 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

the elbow and silhouette methods to study in panel data. By grouping farms based on yield and ares of production, the study proposes to find the best clustering model. The research uses dataset with two variables for a detailed analysis. By using the clustering methods, it may take targeted actions to increase production and support sustainable farming practices.

II. Methodology

A. Data Collection

The data for this study is sourced from the data center in Chanthaburi, Thailand. It includes various attributes related to durian yield and area of production. The variables of the dataset are the year of data collection, the province and district where the data was collected, the area of production in hectare and the yield of durian in kilograms per hectare. The dataset is valuable for providing actionable insights to enhance the productivity and sustainability of durian farming in Eastern Thailand. Therefore, the provided dataset contains detailed information on durian cultivation in various districts across six provinces in eastern Thailand, namely Nakhon Nayok, Prachin Buri, Chonburi, Trat, Chanthaburi and Rayong from 2012 to 2023.

B. Data Analysis

1) K-Means Clustering Algorithm: The k-means clustering algorithm partitions n data points into k clusters where each data point belongs to the cluster with the nearest mean. The objective is to minimize the within cluster sum of squares (WCSS), which is also known as the inertia. The algorithm can be summarized as follows

1) Initialize k cluster centroids randomly.
2) Assign each data point to the nearest cluster centroid.
3) Update the centroids by calculating the mean of all data points assigned to each cluster.
4) Repeat steps 2 and 3 until convergence.

The objective function to be minimized is defined as

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{1}$$

where k is the number of clusters, C_i is the set of points that belong to cluster i and μ_i is the centroid of cluster i, calculated as

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{2}$$

where $\|x - \mu_i\|^2$ is the Euclidean distance between point x and the centroid μ_i.

2) Elbow Method for Evaluating Cluster Quality: The elbow method is a technique used to determine the optimal number of clusters in a dataset by plotting the within cluster sum of squares (WCSS) against the number of clusters (k). The elbow point on the plot indicates the optimal number of clusters, where the rate of decrease in WCSS sharply slows down [15].

To calculate the WCSS for a given number of clusters, there are the two steps for this method. First, run the k-means algorithm for different values of k (from 1 to 10). For each value of k, compute the WCSS, which is the sum of squared distances between each point and its assigned cluster centroid. The WCSS is calculated as

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{3}$$

where k is the number of clusters, C_i is the set of points assigned to cluster i, μ_i is the centroid of cluster i, and $\|x - \mu_i\|^2$ is the Euclidean distance between point x and centroid μ_i.

3) Silhouette Method for Evaluating Cluster Quality: The silhouette method is used to evaluate the quality of clusters by measuring how similar each point is to its own cluster compared to other clusters [16]. The silhouette score ranges from -1 to 1 where a higher value indicates better defined clusters. To calculate the silhouette score that is to calculate the average distance between a point and all other points in the same cluster (a), calculate the average distance between a point and all points in the nearest cluster (b) and compute the silhouette score for each point as follows

$$s = \frac{b - a}{\max(a, b)} \tag{4}$$

where s is the silhouette score, a is the average intra cluster distance and b is the average nearest cluster distance.

C. Performance Metrics for Clustering Evaluation

Evaluating the performance of clustering algorithms is essential to ensure the quality and effectiveness of the clustering results. The performance metrics used to evaluate clustering algorithms, including Davies-Bouldin Index and Dunn Index.

1) Davies-Bouldin Index: The Davies-Bouldin Index (DBI) is a metric used to evaluate the average similarity ratio of each cluster with its most similar cluster [17]. Lower values indicate better clustering. The DBI is calculated as follows

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d_{ij}} \right) \tag{5}$$

where k is the number of clusters, σ_i and σ_j are the average distances between each point in cluster i and j to their respective centroids and d_{ij} is the distance between the centroids of clusters i and j.

2) Dunn Index: The Dunn Index is used to identify dense and well separated clusters. Higher values of the Dunn Index indicate better clustering [18]. The Dunn Index is calculated as follows

$$D = \frac{\min_{1 \leq i < j \leq k} d(C_i, C_j)}{\max_{1 \leq i \leq k} \delta(C_i)} \tag{6}$$

where k is the number of clusters, $d(C_i, C_j)$ is the distance between clusters C_i and C_j and $\delta(C_i)$ is the diameter of cluster

turnitin Page 142 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

C_i defined as the maximum distance between any two points within the cluster.

III. Results

Firstly, this visualization helps in understanding how durian farming has evolved over time in terms of both land usage and productivity providing valuable insights for further analysis.

A. Visualization of Data

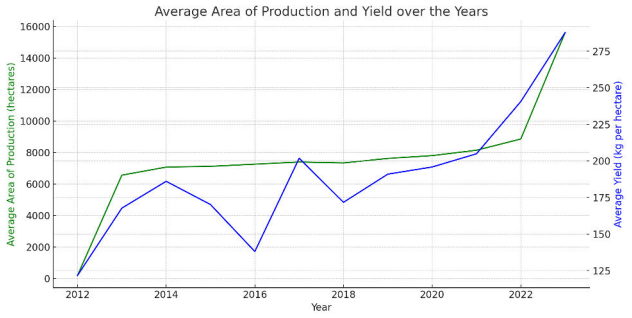


Fig. 1. Average Area of Production and Yield of Durian Over the Years in Eastern Thailand (2012-2023).

Figure 1 shows the trends in the average area of production and yield of durian over the years from 2012 to 2023 in Eastern Thailand. The green line represents the average area of production (in hectares) while the blue line represents the average yield (in kilograms per hectare). Both the average area of production and the average yield of durian have shown a general upward trend over the years. The yield per hectare has fluctuated over the years but has generally increased, particularly from 2018 onwards. It can be seen that the yield line is above the production area line for the years 2021-2023.

B. Determining Optimal Number of Clusters

Figure 2 shows the elbow and silhouette methods applied to the dataset from 2012 to 2023. The x axis represents the number of clusters (k) and the y axis shows the sum of squared distances (SSD) from each point to its assigned cluster center. The plot identifies the elbow point where the reduction in SSD starts to slow down indicating the optimal number of clusters. In addition, figure 2 presents the silhouette scores for various cluster configurations over the same period. The x axis represents the silhouette score which measures how similar an object is to its own cluster compared to other clusters and the y axis shows the number of clusters. Higher silhouette scores indicate better defined and more cohesive clusters.

C. Performance Metrics

The clustering results are essential to consider two evaluation metrics such as the Davies-Bouldin Index and Dunn Index. They provide quantitative measures of clustering quality. A lower Davies-Bouldin Index indicates better defined clusters while a higher Dunn Index suggests well separated and compact clusters. Additionally, visually inspecting the scatter plots helps ensure that the clusters are well separated. The centroids are logically positioned within the clusters rather than being influenced by outliers.

Table I presents various clustering evaluation metrics for the years 2012 to 2023 using two different methods. The metrics include the Davies-Bouldin index and Dunn index. The number of clusters varies by year and method. For instance, in 2012, the recommendation method suggests 2 clusters with a lower Davies-Bouldin index and a higher Dunn index compared to the clustering with elbow method and the clustering with silhouette method. Therefore, for the optimal number of clusters each year based on the evaluation metrics from the elbow and silhouette methods, the recommendations are as follows: In 2012, 2 clusters in 2013, 3 clusters in 2014, 3 clusters in 2015, 5 clusters in 2016, 3 and 4 clusters in 2017, 3 clusters in 2018, 3 clusters in 2019, 4 clusters in 2020, 4 clusters in 2021, 3 clusters in 2022 and 4 clusters in 2023.

Figure 3 is a scatter plot showing the results of k-means clustering for the year 2012, 2021, 2022 and 2023 with 2, 4, 3 and 4 centroids, respectively. The x axis is labeled area of production and the y axis is labeled yield. For the year 2012, the data points are represented by multiple black circular markers, mostly scattered in the lower left corner of the plot indicating lower values of area of production but varying yield values. Two red star shaped centroids are plotted on the graph. One centroid is located near the cluster of data points, approximately at coordinates (x = 50, y = 120) while the other centroid is far from the cluster of data points, approximately at coordinates (x = 1500, y = 120). The plot suggests that the data points form a tight cluster with one centroid close to them while the other centroid is isolated far from the main cluster. This may indicate an imbalance and outlier in the data when using 2 centroids for clustering.

Additionally, there are some data points spread across higher values of the area of production in 2021. Four red star shaped centroids are plotted on the graph each representing the center of a cluster. One centroid is located near the dense cluster of data points (approximately at coordinates (x = 5000, y = 150)) while the other three centroids are positioned among the more dispersed data points (approximately at coordinates (x = 10000, y = 300), (x = 30000, y = 300) and (x = 60000, y = 250)). The legend in the upper right corner of the plot indicates that the red star shaped points represent centroids. The plot suggests that the data points form several distinct clusters, with centroids located at varying levels of area of production and yield. This distribution indicates a more complex clustering pattern compared to the previous year's plot with 2 centroids, highlighting the presence of multiple groupings within the data.

Moreover, in 2022, one centroid is located near the dense cluster of data points (approximately at coordinates (x = 5000, y = 200)) while the other two centroids are positioned among the more dispersed data points (approximately at coordinates (x = 20000, y = 350) and (x = 60000, y = 300)). For 2023, one centroid is located near the dense cluster of data points (approximately at coordinates (x = 10000, y = 250)) while the other three centroids are positioned among the more dispersed

turnitin Page 143 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

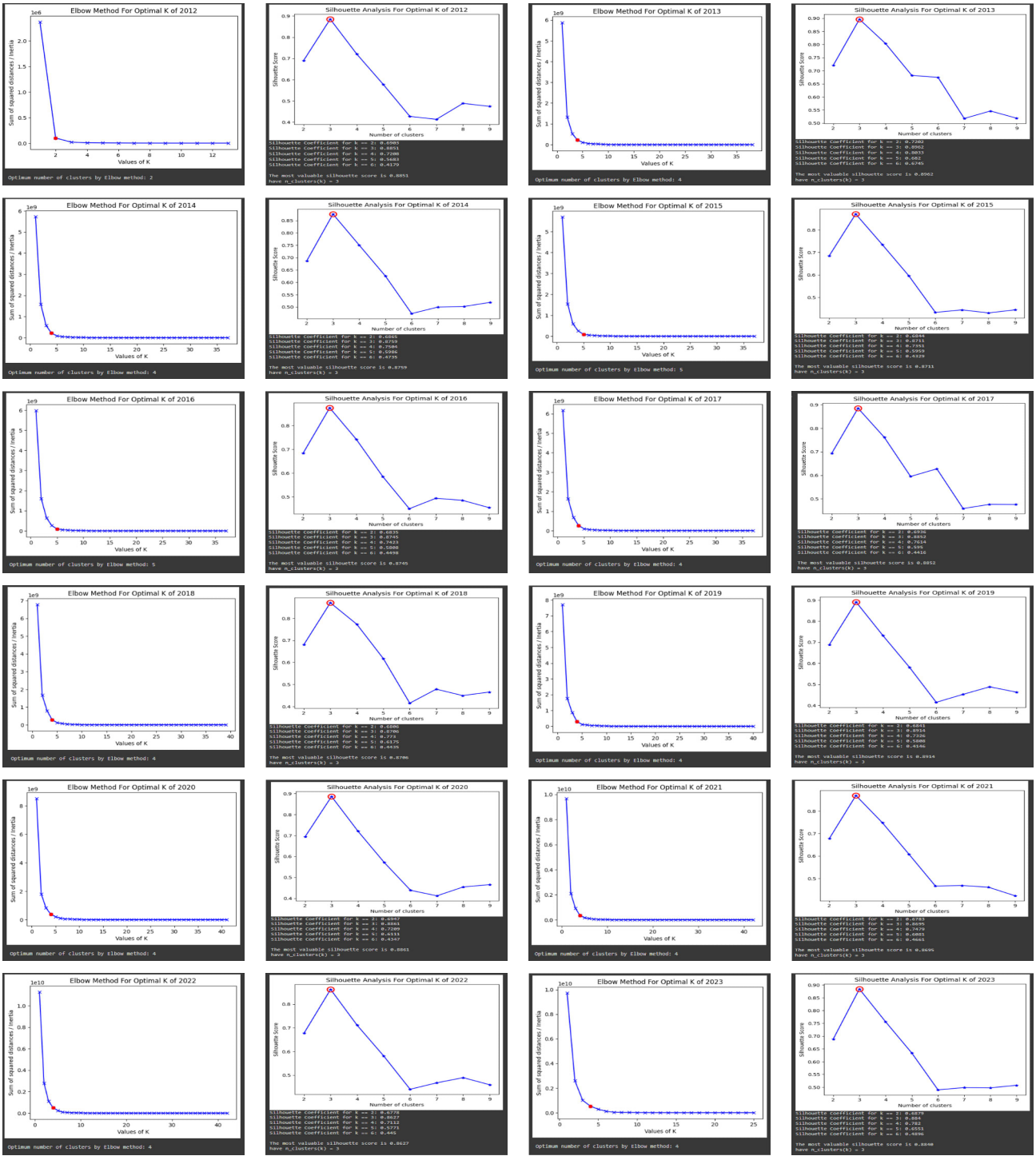


Fig. 2. Elbow and Silhouette Method for Optimal Cluster Determination (2012-2023)

turnitin Page 144 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

TABLE I
CLUSTERING EVALUATION METRICS AND RECOMMENDATIONS

Year	Clustering with Elbow Method			Clustering with Silhouette Method			Recommended Clusters
	Cluster	Davies-Bouldin Index	Dunn Index	Cluster	Davies-Bouldin Index	Dunn Index	
2012	2	0.0416	3.7723	3	0.0964	1.9003	2
2013	4	0.3360	0.1179	3	0.2606	0.4759	3
2014	4	0.3349	0.1559	3	0.2921	0.2833	3
2015	5	0.3143	0.3782	3	0.3373	0.1920	5
2016	5	0.3166	0.3436	3	0.3486	0.1701	5
2017	4	0.3515	0.1337	3	0.3528	0.1557	3,4
2018	4	0.3301	0.1440	3	0.3172	0.2325	3
2019	4	0.3069	0.0812	3	0.4474	0.0382	4
2020	4	0.3523	0.0973	3	0.4220	0.0544	4
2021	4	0.4103	0.1471	3	0.4415	0.0698	4
2022	4	0.4602	0.1378	3	0.4156	0.2596	3
2023	4	0.3952	0.2768	3	0.4479	0.1984	4

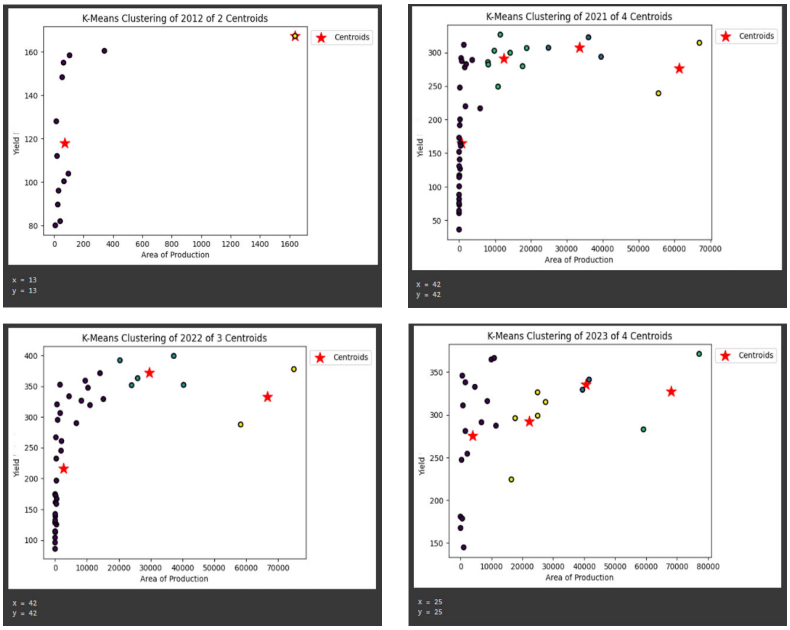


Fig. 3. K-Means clustering analysis of yield and area production data for 2012, 2021, 2022 and 2023

data points (approximately at coordinates (x = 20000, y = 300), (x = 40000, y = 350) and (x = 70000, y = 300)).

IV. DISCUSSION

The clustering analysis conduct for the years 2012, 2021, 2022 and 2023 demonstrates the dynamic nature of durian farming in Eastern Thailand and how data driven methods can enhance agricultural efficiency. The k-means clustering algorithm, applied with the elbow and silhouette methods, has revealed significant insights into the area and productivity patterns of durian farms. For instance, in 2012, the clustering indicate two main clusters with one centroid positioned near a dense group of data points and another isolated centroid suggesting an imbalance and the presence of outliers [19]. This might reflect the variability in farming practices and environmental conditions that year. In 2021, the clustering analysis reveal a more complex pattern with four centroids [20] indicating a diverse range of farming areas and production levels. The centroids are spread across different production areas highlighting the varied agricultural practices and possibly the adoption of more sophisticated farming techniques.

In 2022 and 2023, the clustering results continued to show distinct groupings with centroids reflecting different levels of yield and production areas. The changes in centroid positions over these years suggest an evolution in farming practices possibly influenced by technological advancements and better farm management. For example, it can be seen that Khlung and Thamai in Chanthaburi account for the majority of the production area in Eastern Thailand within one cluster in 2022 and 2023. The performance metrics including the Davies-Bouldin Index and Dunn Index supported these findings by indicating the quality and separability of the clusters [21], [22]. For most years, the elbow method provided better clustering quality metrics but in some instances, the silhouette method indicated superior clustering quality. These insights can be critical for farmers to make informed decisions, optimize

turnitin Page 145 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

resource allocation and implement targeted interventions to enhance productivity and sustainability in durian farming. By leveraging these data driven techniques, the study underscores the potential of precision agriculture in transforming traditional farming practices and boosting economic growth in the region.

V. Conclusion

The study demonstrates the effectiveness of using k-means clustering combined with the elbow and silhouette methods to analyze durian farm yields and production areas in Eastern Thailand from 2012 to 2023. The clustering analysis reveal significant patterns in durian farming showing how different clusters of production areas and yields can be identified. The evaluation metrics, including the Davies-Bouldin Index and Dunn Index, indicate that the elbow method generally provided better defined clusters. Although in some cases, the silhouette method showed superior clustering quality. Future studies will focus on identifying which specific provinces fall into each cluster group providing even more granular insights to further optimize regional farming strategies.

Acknowledgment

Thank you to the Chanthaburi Data Center for supporting agricultural data collection in Eastern Thailand.

References

[1] P. Thongnim, V. Yuvanatemiya, P. Srinil, and T. Phukseng, "An exploration of emission data visualization in southeast asian countries," in *2023 10th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2023, pp. 160–165.

[2] P. Thongnim, V. Yuvanatemiya, and P. Srinil, "Smart agriculture: Transforming agriculture with technology," in *Asia Simulation Conference*. Springer, 2023, pp. 362–376.

[3] Z. Guido, A. Zimmer, S. Lopus, C. Hannah, D. Gower, K. Waldman, N. Krell, J. Sheffield, K. Caylor, and T. Evans, "Farmer forecasts: Impacts of seasonal rainfall expectations on agricultural decision-making in sub-saharan africa," *Climate Risk Management*, vol. 30, p. 100247, 2020.

[4] H. Williams, T. Colombi, and T. Keller, "The influence of soil management on soil health: An on-farm study in southern sweden," *Geoderma*, vol. 360, p. 114010, 2020.

[5] S. Somsri, "Durian: Southeast asia's king of fruits," *Chronica Horticulturae*, vol. 48, no. 4, pp. 19–22, 2008.

[6] C. J. Thorogood, M. N. Ghazalli, M. Y. Siti-Munirah, D. Nikong, Y. W. C. Kusuma, S. Sudarmono, and J. R. Witono, "The king of fruits," *Plants, People, Planet*, vol. 4, no. 6, pp. 538–547, 2022.

[7] N. A. Husin, S. Rahman, R. Karunakaran, and S. J. Bhore, "A review on the nutritional, medicinal, molecular and genome attributes of durian (durio zibethinus l.), the king of fruits in malaysia," *Bioinformation*, vol. 14, no. 6, p. 265, 2018.

[8] P. Tittonell, B. Vanlauwe, P. Leffelaar, E. C. Rowe, and K. E. Giller, "Exploring diversity in soil fertility management of smallholder farms in western kenya: I. heterogeneity at region and farm scale," *Agriculture, ecosystems & environment*, vol. 110, no. 3-4, pp. 149–165, 2005.

[9] P. Srinil and P. Thongnim, "Deep learning enhanced hand gesture recognition for efficient drone use in agriculture." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 5, 2024.

[10] E. Andersen, B. Elbersen, F. Godeschalk, and D. Verhoog, "Farm management indicators and farm typologies as a basis for assessments in a changing policy environment," *Journal of environmental management*, vol. 82, no. 3, pp. 353–362, 2007.

[11] P. Thongnim, E. Charoenwanit, and T. Phukseng, "Cluster quality in agriculture: Assessing gdp and harvest patterns in asia and europe with k-means and silhouette scores," in *2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*. IEEE, 2023, pp. 1–5.

[12] V. K. Gunjan, "Instantaneous approach for evaluating the initial centers in the agricultural databases using k-means clustering algorithm," *Journal of mobile multimedia*, vol. 18, no. 1, pp. 43–60, 2021.

[13] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[14] D. M. Saputra, D. Saputra, and L. D. Oswari, "Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method," in *Sriwijaya international conference on information technology and its applications (SICONIAN 2019)*. Atlantis Press, 2020, pp. 341–346.

[15] M. Cui *et al.*, "Introduction to the k-means clustering algorithm based on the elbow method," *Accounting, Auditing and Finance*, vol. 1, no. 1, pp. 5–8, 2020.

[16] M. Shutaywi and N. N. Kachouie, "Silhouette analysis for performance evaluation in machine learning with applications to clustering," *Entropy*, vol. 23, no. 6, p. 759, 2021.

[17] Y. A. Wijaya, D. A. Kurniady, E. Setyanto, W. S. Tarihoran, D. Rusmana, and R. Rahim, "Davies bouldin index algorithm for optimizing clustering case studies mapping school facilities," *TEM J*, vol. 10, no. 3, pp. 1099–1103, 2021.

[18] S. Mahallati, J. C. Bezdek, D. Kumar, M. R. Popovic, and T. A. Valiante, "Interpreting cluster structure in waveform data with visual assessment and dunn's index," *Frontiers in Computational Intelligence*, pp. 73–101, 2018.

[19] P. J. Jones, M. K. James, M. J. Davies, K. Khunti, M. Catt, T. Yates, A. V. Rowlands, and E. M. Mirkes, "Filterk: A new outlier detection method for k-means clustering of physical activity," *Journal of biomedical informatics*, vol. 104, p. 103397, 2020.

[20] M. M.-T. Chiang and B. Mirkin, "Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads," *Journal of classification*, vol. 27, pp. 3–40, 2010.

[21] F. Ros, R. Riad, and S. Guillaume, "Pdbi: A partitioning davies-bouldin index for clustering evaluation," *Neurocomputing*, vol. 528, pp. 178–199, 2023.

[22] C.-E. B. Ncir, A. Hamza, and W. Bouaguel, "Parallel and scalable dunn index for the validation of big data clusters," *Parallel Computing*, vol. 102, p. 102751, 2021.

turnitin Page 146 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Predicting China's Marriage Rate: Causal Inference Using Dual Machine Learning (DML) with XGBoost, LightGBM, CatBoost, and GBDT

Deyu Zhang
School of Information Technology
Mae Fah Luang University
Chiang Rai, Thailand
6451301504@lamduan.mfu.ac.th

Worasak Rueangsirarak
Computer and Communication Engineering for Capacity Building Research Unit
School of Information Technology
Mae Fah Luang University
Chiang Rai, Thailand
worasak.rue@mfu.ac.th

Surapong Uttama
Center of Excellence in Artificial Intelligence and Emerging Technologies
School of Information Technology
Mae Fah Luang University
Chiang Rai, Thailand
surapong@mfu.ac.th

***Abstract*—After China's accession to the WTO and 20 years of rapid development, the marriage rate has shown a downward trend. This study aims to analyze the impact of socio-economic factors on the crude marriage rate (CMR) panel data in China from 2003 to 2022 using Dual Machine Learning (DML) for Causal Inference and machine learning models. DML was used to estimate ATE, CATE, and HTE for various factors. Four models—XGBoost, LightGBM, CatBoost, and GBDT—were employed for predictions, using 10-fold cross-validation for model evaluation. The results indicate that education (ATE 5.666) and birth rate (ATE 5.492) had the most significant positive impacts on CMR, while GDP (ATE 3.196) showed positive but varying effects, and the female proportion (ATE -2.353) had a notable negative impact. CatBoost performed best in MSE (0.942) and RMSE (0.958), while LightGBM excelled in MAE (0.777). Education, GDP, and birth rate are key factors influencing CMR. CatBoost and LightGBM proved to be effective prediction models, though improvements are needed for regions with significant variability.**

Keywords—Marriage Rate, DML, XGBoost, CatBoost, LightGBM, GBDT, ATE, CATE, RMSE,

I. INTRODUCTION

Marriage plays a crucial role in numerous social phenomena, such as happiness, procreation, child rearing, gender disparities, criminal behavior, and labor force participation. China is known for its culture of widespread marriage, yet the trend of delaying the age of first marriage is gaining prominence [1]. In recent years, There has been an increase in delayed marriage age and declining fertility rates in China, possibly due to high housing prices acting as catalysts. Marriage is a fundamental social institution with significant impacts on various societal aspects such as happiness, reproduction, child development, gender inequality, and crime. It also plays a crucial role in addressing labor supply in the employment sector. However, there has been a sharp decline in marriage rates in many countries, starting in developed nations like Western Europe and the United States, and followed by East Asian countries such as Japan and South Korea, with China closely behind [2]. The issue of marrying in China is complex. The complexity of marriage is not only related to emotions, but more importantly, to economic factors. Among the important factors involved in marriage are elements such as GDP, housing prices, income, consumption, pension ratio, gender ratio, and education level [3]. At the same time, similar problems are also emerging in India, a populous country in Asia [4]. Factors influencing the marriage rate in China include the responsibility of men to purchase marital housing before marriage, high property prices discouraging young people [1].and the savings ability of men and their families [5]. The dual structure of urban and rural areas, with rural and urban household registrations, leads to women being willing to marry into cities. The male population exceeds the female population, contributing to gender imbalance as a key factor [6]. causal machine learning inference pipeline that combines a given predictive machine learning model with analytical estimations of average treatment effects [7].

Accurately predicting marriage rates is crucial for governments to formulate policies and strategic decisions. Simple linear regression is insufficient for addressing current research questions. Time series analysis can study the temporal dynamics within individual entities, while panel data analysis is more suitable for studying changes across multiple entities over time. Controlling for unobserved heterogeneity and conducting comparative analysis are essential. However, due to the limited frequency of our study data, which spans only 20 years, using time series analysis may not yield the expected results.. This makes panel data a better choice for research involving annual data from different regions, countries, or other entities. This study delves into the intricate dynamics of marriage rates, revealing seven key factors through the application of machine learning techniques [8]. The effectiveness of four regression models—XGBoost, LightGBM, CatBoost, GBDT models —is assessed, highlighting the exceptional performance of the CatBoost model and the robust predictive capabilities [10][20][24].

The paper is structured as follows. Section 2 begins with an overview of the previous researches on marriage rate and factors affecting it. Section 3 describes research methodology, Section 4 presents results and discussion and section 5 draws a conclusion.

II. LITERATURE REVIEW

A. Related Studies

In this section, Zhao et al. studied the impact of increasing housing costs on marriage postponement in China. They used the Difference-in-Differences (DID) method to analyze how rising prices affect marriage timing. They discovered that higher prices raise the expenses of initial marriages. Moreover, the delay in marriage due to cost escalation is more noticeable in individuals with advanced female education,

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

more brothers among males, and those from urban areas. This delay in marriage due to cost surge also decreases the inclination for childbirth, resulting in lower fertility rates. [3]. Gaurav Chiplunkar and others studied Marriage markets and the rise of dowry in India, and they found that between 1930 and 1975, the proportion of Indian marriages involving dowry payments doubled, with the average actual value of payments tripling [4]. Author Patrik Guggenberger examines the scale properties of two-stage tests in panel data models, with the first stage employing a Hausman (1978) specification test as a pretest for the random effects specification, and the second stage using a test statistic based on random effects or fixed effects estimates, depending on the result of the Hausman pretest [10]. Ma Pilar Alonso and colleagues conduct a study on financial exclusion, depopulation, and aging using panel data regression models to analyze the influence of social demographic characteristics on financial exclusion. [11]. Shailendra Sharma and other researchers study House Price Prediction with Machine Learning Algorithms, emphasizing precise prediction using Python libraries like matplotlib, pandas, and NumPy. The widely used Python library for machine learning, scikit-learn, is open-source. [12]. Shailendra Sharma et al. developed an efficient method for least squares estimation in dynamic space-time panel data models. This method, called eigendecomposition-based bias-corrected least squares procedure, uses eigen decomposition of the weight matrix in dynamic space-time pooled panel data models. Vita Ratnasari proposed a statistical model to analyze factors influencing the middle-income trap in Indonesia through panel data regression, using observations at the provincial level based on inter-regional decomposed variables. [14].

The main contribution of our study lies in the approach of combining machine learning with panel data and Dual Machine Learning (DML) for Causal Inference for analysis, model evaluation and prediction.Dual Machine Learning (DML) for Causal Inference using XGBoost, LightGBM, CatBoost, GBDT is used. The values of ATE (Average Treatment Effect) and CATE (Conditional Average Treatment Effect) and HTE (Heterogeneous Treatment Effect) of the independent variables on crude marriage are obtained. Four machine learning models are used, namely XGBoost, LightGBM, XGBoost and CatBoost, to train the data from 2003 to 2022. Model evaluation involves MSE, RMSE and MAE metrics. After model evaluation, the four models are used to predict the crude marriage rate in 2022 by comparing the actual crude marriage rate data and the predicted values in 2022. Combining panel-arranged data with Dual Machine Learning (DML) for Causal Inference and four machine learning models can build a more complex and accurate prediction model, providing a new perspective for our in-depth understanding of the changing trend of China's marriage rate.

B. Term definition

According to the United Nations, the crude marriage rate (CMR) is a vital statistics summary rate based on the number of marriages occurring in a population during a given period, usually a calendar year. It is calculated as the number of marriages occurring among the population of a given geographical area during a given year per 1,000 mid-term population of the same area during the same year. The formula for the crude marriage rate(CMR) is (1) [16].

$$CMR = \frac{Number\ of\ Marriage\ Registration\ Pairs}{Mid-Term\ Population} * 1000 \quad (1)$$

C. The Machine Learning Model

Panel data is a combination of cross-sectional data and time series data. Cross-sectional data involves observing multiple entities' variables at a specific point in time, while time series data involves observing a single entity repeatedly over time [15]. Panel data integrates both features by gathering data from identical subjects over time, similar to observing the same individuals at consistent intervals on a timeline [14][18].

1) XGBoost: A gradient boosting based tree model optimized for speed and performance to handle large-scale data [24].

*2) LightGBM:*A fast, distributed gradient boosting framework based on decision tree algorithms, suitable for big data. [24].

3) CatBoost: Gradient boosted decision tree algorithm for categorical features, reducing overfitting and improving predictive performance [24].

4) GBDT: Gradient boosted decision trees, gradually build models to improve prediction accuracy, by weighted combination of weak learners [24].

5) Dual Machine Learning (DML) for Causal Inference: Causal inference was proposed to create interpretable, robust, and powerful machine learning models. Its core approach is to measure cause-effect relationships. It is ubiquitous in decision-making problems in various fields such as healthcare and economics. A machine learning approach for causal inference that combines machine learning models with dual estimation techniques from economics to reduce bias and improve the accuracy of estimates [22][25].

6) ATE (Average Treatment Effect): The mean treatment effect, which measures the average effect of the treatment on the outcome variable in the population, reflects the population-wide causal effect [23]. The formula for the ATE is (2).

$$ATE = E[Y(T = 1) - Y(T = 0)] \quad (2)$$

- where Y(T = 1)|T = 1 and Y(T = 0)|T = 1 are the potential treated and control outcomes of the treated group, respectively. ATT can also be called local average treatment effect (LATE).

7) CATE (Conditional Average Treatment Effect): The conditional mean treatment effect, which measures the average effect of the treatment on the outcome variable under a particular condition, reflects differences in causal effects across subpopulations [23]. The formula for the CATE is (3).

$$CATE = E[Y(T = 1)|X = x] - E[Y(T = 0)|X = x] \quad (3)$$

- where Y(T = 1)|X = x and Y(T = 0)|X are the potential treated and control outcomes of the subgroup with X = x respectively. CATE is also known as the heterogeneous treatment effect.

8) HTE (Heterogeneous Treatment Effect): HTE refers to the variation in the impact of a treatment across different individuals or groups. In simpler terms, it means that the same treatment can have different effects on different people. General Form[23]. The formula for the *HTE* is (4),(5).

$$Y_i = \alpha + \beta T_i + \gamma X_i + \epsilon_i \quad (4)$$

- where Y_i is the outcome variable (e.g., treatment effect), T_i is the treatment indicator variable (1 indicates treatment received, 0 indicates no treatment), X_i represents individual characteristics (covariates), α, β, and γ are parameters to be estimated, ϵ_i is the error term.

$$y_i = \alpha + \beta T_i + \delta(T_i * X_i) + \epsilon_i \quad (5)$$

- where, $\delta\delta$ represents the effect of the interaction between treatment effect and covariates, reflecting the heterogeneity of treatment effects.

III. Research Methodology

A. Overall Methodology

In this section, we define the issue of marriage rate in China, collect data, focus on independent variables, and conduct quantitative analysis. Missing data is handled in the preprocessing step. We propose a mixed model for marriage rate prediction using three models based on panel data and machine learning for evaluation. Use Dual Machine Learning (DML) for Causal Inference using XGBoost, LightGBM, CatBoost, GBDT to train machine learning models.we finally evaluate model performance and predictions with MSE,RMSE and MAE. [10][14]. We focus on the most critical characteristics that have a significant impact on the crude marriage rate, such as distance from GDP, house price, gross dependency ratio, Birth Rate, Female, Average years of education, and Sex Ratio. As shown in Fig.1, shows the Overall Methodology of our research paper, focusing on crude marriage rates using panel data analysis and Machine learning model evaluation and prediction.

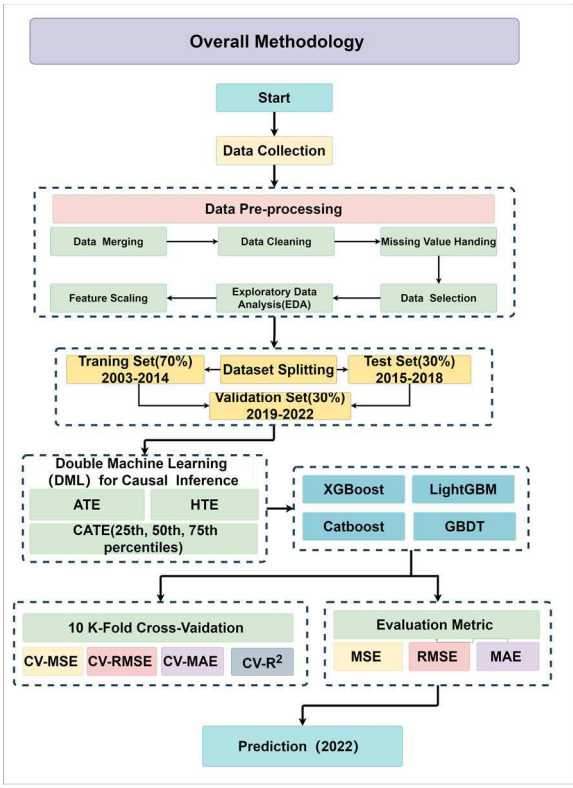


Fig. 1. Overall Methodology

B. Data Collection

In this section, thorough explanations of the key variables impacting the crude marriage rate are presented, encompassing data collection and processing. The crude marriage rate and economic factors data are obtained from the National Bureau of Statistics of China (https://data.stats.gov.cn/english/) and the China Statistical Yearbook (https://www.stats.gov.cn/sj/ndsj/). Each relevant dataset undergoes annual updates on the National Bureau of Statistics of China's website and encompasses the years 2003 to 2022, culminating in a 20-year dataset.

C. Data Merging

In the study, due to the independence of the datasets provided by the National Bureau of Statistics of China and the China Statistical Yearbook, there is no established correlation between each dataset. Additionally, the relevant independent variables and dependent variables available for download on the public platform consist of several separate Excel spreadsheets. The maximum span of publicly available datasets is 20 years, specifically from 2003 to 2022. To facilitate future research due to the multitude of tables, we consolidated the useful data for the study, creating distinct Excel and CSV format files. The columns in the CSV file data include Region, Year, Dependent Variables, and Independent Variables.

D. Data Pre-Processing

Scikit-learn, a Python library for machine learning, was utilized to combine and preprocess data from various independent CSV files in the research. The missing values in certain columns were handled through the Simple Imputer tool available in the SciPy library, ensuring the integrity of the data [15].

E. Data Selection

After merging and preprocessing the data, we decided to select the data column features to include the 31 provinces in mainland China, the years are 2003-2022, the dependent variable y is Crude marriage rate, and the independent variables are (X_1-X_7). Table Ⅰ offers an explanation of the arrangement of the newly screened features.

TABLE I. Features Date Selection

Features	Features Explanation
Region	31 provinces in mainland China（No data from Hong Kong, Macao and Taiwan）
Year	2003-2022
Crude_marriage_rate（Y）	The Number of Marriages 0ccurring in a Population During a Given Period
GDP (X_1)	Gross Regional Product (100 million yuan)
House_ Prices (X_2)	Average Selling Price of Commercialized Residential Buildings (yuan / square meters)
Gross_Dependency_Ratio (X_3)	Gross Dependency Ratio (Sample Survey) (%)
Birth_Rate (X_4)	Birth Rate (%)
Female (X_5)	Female Population Aged 15 and Over (Sample Survey) (person)
Average_years_of_education (X_6)	Average years of education per capita
Sex_Ratio (X_7)	Sex Ratio (Female=100) (Sample Survey) (female=100)

F. Feature Selection

As shown in Fig.2, there is a heat map of a crude marriage rate dataset. It is evident that, except between GDP and Birth Rate, House Prices and Gross Dependency Ratio, between Birth Rate, Average years of education and Gross Dependency Ratio, and between Average years of education and Birth Rate, are negatively correlated, while all others are positively correlated.

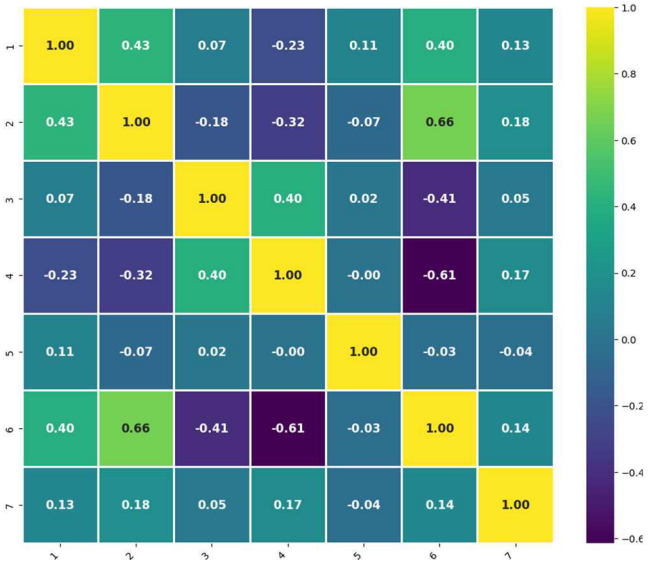


Fig. 2. Correlation Heatmap of Features

G. Feature Scaling

Data normalization, known as feature scaling, is a key preprocessing step in many regression-oriented machine learning models. It involves standardizing numerical attributes to a common scale. In this study, Min-Max Scaling was used to normalize attributes to a range of 0 to 1, aiming to reduce the impact of dimension variations and improve model training efficiency and reliability [15].

H. Evaluation Model

The regression model evaluation indicators RMSE, MAE, indicators are mainly used to evaluate the prediction error rate and model performance in regression analysis [19].

1) Mean squared error (MSE): Mean squared error (MSE) is a common measure of the quality of an estimator, such as a machine learning model. It calculates the average squared difference between the predicted values and the actual values. A lower MSE value indicates a better fit of the model to the data [19]. The formula for the RMSE is (6).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{6}$$

*2) Root mean squared error (RMSE):*Root mean squared error (RMSE) is the square root of the mean squared error (MSE). It is another common measure of the quality of an estimator, and it represents the average error in the predictions [19]. The formula for the RMSE is (7).

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y})^2} \tag{7}$$

- Where N is the number of samples, Y_i is the true value, $\hat{y}$ is the predicted value

3) Mean absolute error (MAE): Mean absolute error (MAE) measures estimator quality by averaging absolute differences between predicted and actual values.A lower MAE value indicates a better fit of the model to the data [19]. The formula for the MAE is (8).

$$MAE = \frac{1}{N}\Sigma_{i=1}^{N}|y_i - \hat{y}| \tag{8}$$

4) K-Fold Cross Validation : 10-K-Fold Cross Validation is a commonly used model evaluation method to estimate the performance of machine learning models on unseen data. The first step is data partitioning, which randomly divides the data set into 10 equal parts (called "folds"). The second step is model training and validation, which is performed 10 times. One fold is used as the validation set and nine folds are used as the training set. The third step is to calculate the average performance: the evaluation results of the 10 validations are averaged as the final performance indicator of the model [22]. The formula for the 10-K-Fold Cross Validation is (9).

$$L_{cv} = \frac{1}{10}\sum_{k=1}^{10} L^k \tag{9}$$

- Where L_{cv} is represents the average loss (or error) across all 10 folds in the cross-validation process.
- L^k is denotes the loss (or error) calculated for the *k*-th fold during the cross-validation.

IV. Results and Discussion

A. Results of ATE and CATE for Independent Variable Features

Table III illustrates the impact of various independent variables on the Crude Marriage Rate (CMR), analyzed using Double Machine Learning (DML) to estimate the Average Treatment Effect (ATE), Conditional Average Treatment Effect (CATE), and Heterogeneous Treatment Effect (HTE). Notably, the Birth Rate (X_4) and Average Years of Education (X_6) exhibit the highest ATE values of 5.492 and 5.666, respectively, indicating a strong positive influence on CMR. In contrast, the Female (X_5) variable shows a significant negative ATE (-2.353), suggesting a decrease in CMR. The HTE values reveal variability in the effects; for example, GDP (X_1) shows substantial heterogeneity (4.968), implying varied impacts across different subpopulations. House Prices (X_2) has a negative HTE of -10.145, indicating a consistent negative effect on CMR across different groups. The CATE values at the 25th, 50th, and 75th percentiles highlight the conditional effects, with notable variations observed in House Prices (X_2) and Birth Rate (X_4), suggesting context-specific influences on CMR. These findings underscore the nuanced and multifaceted relationships between these variables and the Crude Marriage Rate.

TABLE II. THE VALUE OF THE INDEPENDENT VARIABLES ESTIMATED BY DML ON ATE, CATE AND OF CRUDE MARRIAGE RATE

Feature	The Results of Features, ATE, CATE and HTE				
	ATE	CATE (25th, 50th, 75th percentiles)			HTE
		25th	*50th*	*75th*	
GDP (X_1)	3.196	0.000	0.331	-1.065	4.968
House_ Prices (X_2)	-0.443	0.000	-1.336	1.222	-10.145
Gross_Dependency_Ratio (X_3)	0.352	0.626	-0.962	0.737	0.675
Birth_Rate (X_4)	5.492	1.211	0.097	1.165	4.932
Female (X_5)	-2.353	0.000	0.000	-0.708	-3.314
Average_years_of_education (X_6)	5.666	0.456	-0.163	-0.883	5.544
Sex_Ratio (X_7)	-0.706	-1.041	0.029	-0.184	-0.619

B. K-Fold Cross Validation Results

The table compares four models: XGBoost, LightGBM, CatBoost and GBDT, The Table IV presents the results of four different regression models: 10 K-Fold Cross Validation Results for 4 Models are compared based on their R-squared values.

1) 10 K-Fold Cross Validation Results: Table III shows the performance comparison of four machine learning models in 10-fold cross validation. In the 10-fold cross validation, the performance of the four models, XGBoost, LightGBM, CatBoost, and GBDT, were compared. The results show that CatBoost performs best in all evaluation indicators, with an MSE of 0.942, an RMSE of 0.958, a MAE of 0.704, and an R^2 of 0.780. GBDT follows closely with an MSE of 0.956, an RMSE of 0.963, a MAE of 0.763, and an R^2 of 0.774. XGBoost also shows strong performance with an MSE of 0.979, an RMSE of 0.975, a MAE of 0.746, and an R^2 of 0.771. LightGBM has the highest MSE and RMSE values, at 1.060 and 1.021 respectively, a MAE of 0.777, and the lowest R^2 at 0.749. In summary, CatBoost outperforms the other three models in terms of prediction accuracy, especially in the MAE and R^2 indicators.

TABLE III. 10 K-FOLD CROSS VALIDATION RESULTS FOR 4 MODELS

Evaluation Metric	10 K-Fold Cross Validation Results for 4 Models			
	XGBoost	*LightGBM*	*CatBoost*	*GBDT*
CV-MSE	0.979	1.060	0.942	0.956
CV-RMSE	0.975	1.021	0.958	0.963
CV-MAE	0.746	0.777	0.704	0.763
CV-R^2	0.771	0.749	0.780	0.774

C. Prediction of Marriage Rate

The study shows that Table IV lists the results of four different regression models. As shown in Figure 4, the performance indicators of the four models are compared. As shown in Fig.3, the actual marriage rate and predicted marriage rate in 2022. Detailed analysis of the study below.

1) Model Results: Table V shows the evaluation index results of the four machine learning models. In the model evaluation, the performance of XGBoost, LightGBM, CatBoost, and GBDT are compared by MSE, RMSE, and MAE. The results show that CatBoost performs best in MSE and RMSE indicators, which are 3.534 and 1.880 respectively, but is slightly inferior to LightGBM in MAE. LightGBM has an MSE of 3.862 and an RMSE of 1.965, but performs best in the MAE indicator with a value of 1.536. XGBoost's MSE and RMSE are 4.070 and 2.017 respectively, slightly higher than those of CatBoost and LightGBM, with an MAE of 1.634. GBDT has the worst performance, with an MSE of 4.425, an RMSE of 2.103, and an MAE of 1.689. Overall, CatBoost performs best in MSE and RMSE, while LightGBM performs well in MAE.

2) Prediction: This Fig.3 shows the compares the actual and predicted crude marriage rates across different regions in 2022, using four machine learning models: XGBoost, LightGBM, CatBoost, and GBDT. The solid lines depict actual marriage rates, while the dotted lines represent the models' predictions. Overall, the actual rates exhibit significant regional fluctuations, while the models' predictions follow a more consistent trend. For instance, in Beijing, XGBoost closely matches the actual values, while LightGBM is lower, and CatBoost is higher. In Hebei, XGBoost and CatBoost are accurate, whereas LightGBM and GBDT show deviations. In Inner Mongolia, LightGBM performs better, while XGBoost and CatBoost underpredict. In Shanxi, all models overpredict. Liaoning shows higher predictions from all models, contrasting with the lower actual rate. This overprediction pattern is also seen in Hainan and Guangxi, while Heilongjiang and Jiangsu are underpredicted. These results highlight the need for improvement, especially in regions with high variability in marriage rates, and provide insights for selecting more accurate prediction tools.

TABLE IV. 4 MODEL EVALUATION INDEX RESULTS

Evaluation Metric	The Summary of Model Results			
	XGBoost	*LightGBM*	*CatBoost*	*GBDT*
MSE	4.070	3.862	3.534	4.425
RMSE	2.017	1.965	1.880	2.103
MAE	1.634	1.536	1.539	1.689

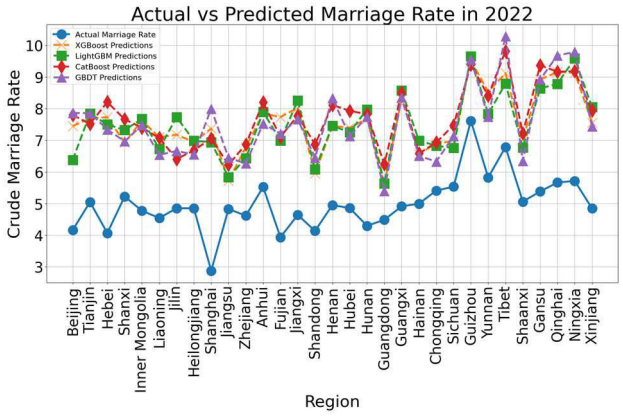


Fig. 3. Actual vs Predicted Marriage Rate in 2022

D. Discussion

In the discussion section, The study found that education (ATE 5.666), GDP (ATE 3.196), and birth rate (ATE 5.492)

are primary factors positively impacting the crude marriage rate (CMR) in China, while the female proportion (ATE -2.353) and house prices (ATE -0.443) have negative effects. These results align with prior research emphasizing the positive influence of education and GDP on marriage rates, though the negative impact of the female proportion diverges from some studies. High ATEs for education and birth rate underscore their crucial roles in boosting CMR, while variable CATE values for GDP (0.000 to -1.065) indicate its fluctuating impact across different quantiles. Policymakers should focus on enhancing education and managing economic conditions to improve marriage rates. Addressing gender disparities is also important. The study's reliance on data from 2003-2022 may not capture recent socio-economic changes, and prediction errors in high-variability regions limit the findings. Future research should include more recent data and additional variables, aiming to improve model accuracy, especially in regions with significant marriage rate variability.

V. Conclusion

This study aimed to analyze the impact of various socio-economic factors on the crude marriage rate (CMR) in China using DML and machine learning models. Key findings include significant positive impacts of education (ATE 5.666) and birth rate (ATE 5.492) on CMR, and a notable negative impact of the female proportion (ATE -2.353). GDP (ATE 3.196) showed positive but variable effects across different quantiles. This research provides a comprehensive analysis of CMR determinants using advanced DML techniques and offers a performance comparison of four machine learning models (XGBoost, LightGBM, CatBoost, GBDT). CatBoost outperformed in terms of MSE (0.942) and RMSE (0.958), while LightGBM had the best MAE (0.777). Future studies should focus on incorporating more recent data and additional variables to improve model performance. Special attention should be given to regions with high variability in marriage rates to enhance predictive accuracy.

Acknowledgment

This research was supported by a grant from Mae Fah Luang University.

References

[1] D. H. Wrenn, J. Yi, and B. Zhang, “House prices and marriage entry in China,” Regional Science and Urban Economics, vol. 74, pp. 118–130, Jan.2019

[2] J. C. Yong, N. P. Li, P. K. Jonason, and Y. W. Tan, “East Asian low marriage and birth rates: The role of life history strategy, culture, and social status affordance,” *Personality and Individual Differences*, vol. 141, pp. 127–132, Apr. 2019

[3] C. Zhao, B. Chen, and X. Li, “Rising housing prices and marriage delays in China: Evidence from the urban land transaction policy,” *Cities*, vol. 135, p. 104214, Apr. 2023

[4] G. Chiplunkar and J. Weaver, “Marriage markets and the rise of dowry in India,” *Journal of Development Economics*, vol. 164, p. 103115, Sep. 2023

[5] G. Nie, “Marriage squeeze, marriage age and the household savings rate in China,” *Journal of Development Economics*, vol. 147, p. 102558,Nov.2020.

[6] J. Chen and W. Pan, “Bride price and gender role in rural China,” *Heliyon*, vol. 9, no. 1, p. e12789, Jan. 2023

[7] J. Rust and S. Autexier, "Causal Inference for Personalized Treatment Effect Estimation for given Machine Learning Models," 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), Nassau, Bahamas, 2022, pp. 1289-1295, doi: 10.1109/ICMLA55696.2022.00206.

[8] Chaturvedi, A. (2023). Dynamic space–time panel data models: An eigendecomposition-based bias-corrected least squares procedure. *Spatial Statistics*, *56*, 100758.

[9] D. Xiao-zhu and K. Ling-wei, "The land prices and housing prices — Empirical research based on panel data of 11 provinces and municipalities in Eastern China," 2013 International Conference on Management Science and Engineering 20th Annual Conference Proceedings, Harbin, China, 2013, pp. 2118-2123

[10] Guggenberger, P. (2010). The impact of a Hausman pretest on the size of a hypothesis test: The panel data case. *Journal of Econometrics*, *156*(2), 337–343.

[11] S. Khan, “Female education and marriage in Pakistan: The role of financial shocks and marital customs,” *World Development*, vol. 173, p.106413,Jan.2024.

[12] S. Sharma, D. Arora, G. Shankar, P. Sharma and V. Motwani, "House Price Prediction using Machine Learning Algorithm," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 982-986

[13] Bresson, G., & Chaturvedi, A. (2023). Dynamic space–time panel data models: An eigendecomposition-based bias-corrected least squares procedure. *Spatial Statistics*, *56*, 100758.

[14] Ratnasari, V., Audha, S. H., & Dani, A. T. R. (2023). Statistical modeling to analyze factors affecting the middle-income trap in Indonesia using panel data regression.

[15] A. Chaurasia and I. U. Haq, "Housing Price Prediction Model Using Machine Learning," 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, 2023, pp. 497-500,

[16] United Nations. “Crude Marriage Rate 1 definition,” https://data.un.org/Glossary.aspx?q=crude+marriage+rate+,2001.

[17] Shanghai Survey Team of National Bureau of Statistics, “Average years of education per capita,” https://tjj.sh.gov.cn/zcjd/20091102/0014-86153.html, November 2009.

[18] De Iaco, S., Palma, M., & Posa, D. (2015). Spatio-temporal geostatistical modeling for French fertility predictions. *Spatial Statistics*, *14*, 546–562.

[19] R. Gupta, A. Sharma, V. Anand and S. Gupta, "Automobile Price Prediction using Regression Models," 2022 International Conference on Inventive Computation Technologies (ICICT), Nepal, 2022, pp. 410-416, doi: 10.1109/ICICT54344.2022.9850657.

[20] C. N. Obiora, A. Ali and A. N. Hasan, "Implementing Extreme Gradient Boosting (XGBoost) Algorithm in Predicting Solar Irradiance," 2021 IEEE PES/IAS PowerAfrica, Nairobi, Kenya, 2021, pp. 1-5, doi: 10.1109/PowerAfrica52236.2021.9543159.

[21] M. Wang *et al.*, “An XGBoost-SHAP approach to quantifying morphological impact on urban flooding susceptibility,” *Ecological Indicators*, vol. 156, p. 111137, Dec. 2023, doi: 10.1016/j.ecolind.2023.111137.

[22] A. Kumar, S. Dodda, N. Kamuni and R. K. Arora, "Unveiling the Impact of Macroeconomic Policies: A Double Machine Learning Approach to Analyzing Interest Rate Effects on Financial Markets," 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-6, doi: 10.1109/AIIoT58432.2024.10574726.

[23] Y. Huang et al., "Robust Causal Learning for the Estimation of Average Treatment Effects," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-9, doi: 10.1109/IJCNN55064.2022.9892344.

[24] S. R, S. S. Ayachit, V. Patil and A. Singh, "Competitive Analysis of the Top Gradient Boosting Machine Learning Algorithms," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 191-196, doi: 10.1109/ICACCCN51052.2020.9362840.

[25] K.-H. Cohrs, G. Varando, N. Carvalhais, M. Reichstein, and G. Camps-Valls, “Causal hybrid modeling with double machine learning,” *arXiv.org*, Feb. 20, 2024. https://arxiv.org/abs/2402.13332

Comparison of the Statistical and Autoencoder Approach for Anomaly Detection in Big Data

Barasha Mali
Department of Electrical and Instrumentation Engineering
Sant Longowal Institute of Engineering and Technology
Longowal, Sangrur, India
barashamali@sliet.ac.in

***Abstract*—This paper compares two anomaly detection methods, comparing the Z-score statistical technique with autoencoders for big datasets, which are crucial for industries like manufacturing, energy, and transportation to maintain smooth operations and avoid costly disruptions. Autoencoders outperformed Z-score statistical technique in anomaly detection on big datasets, achieving higher precision (0.94), F1-score (0.97), and recall (1.00) compared to Z-score statistical technique. This highlights autoencoders' superior ability to accurately identify anomalies, making them more effective for robust anomaly detection in complex data environments.**

***Index Terms*—anomaly, big data, statistical techniques, machine learning, autoencoders**

I. Introduction

Anomaly detection in big data streams of industrial data is important in today's scenario because it helps industrial sectors like manufacturing, energy, and transportation maintain smooth operations and prevent costly disruptions. By continuously monitoring data from machines, sensors, and processes in real time, anomaly detection can swiftly identify deviations that may indicate equipment failures, production inefficiencies, or safety hazards [1], [2].

In industries where downtime can lead to significant financial losses and safety risks, early detection of anomalies is essential. It allows maintenance teams to intervene promptly, minimizing downtime and preventing potential accidents. Moreover, proactive anomaly detection supports predictive maintenance strategies, where equipment issues are addressed before they escalate, optimizing productivity and extending the lifespan of critical assets.

With advancements in technology, such as IoT sensors and advanced analytics, industrial sectors can now leverage sophisticated anomaly detection algorithms to monitor complex data streams effectively. This capability not only enhances operational efficiency but also strengthens overall reliability and resilience, ensuring that industrial processes remain competitive and compliant with stringent safety and regulatory standards in today's competitive market environment [3], [4].

Anomaly detection methods can be categorized into several groups based on their underlying techniques and approaches. Statistical methods rely on measures like mean, standard deviation, or percentiles to identify data points that significantly differ from the norm. Machine learning techniques encompass supervised, unsupervised, and semi-supervised approaches, where algorithms are trained on labeled or unlabeled data to distinguish between normal and anomalous instances. Time series methods analyze sequential data to detect deviations over time, employing techniques such as ARIMA models or moving averages. Furthermore, there are graph-based methods that model relationships between data points as graphs to identify unexpected connections or structural changes indicative of anomalies. Hybrid approaches combine multiple methods to leverage their strengths and improve detection accuracy, adapting to the specific characteristics and challenges of the data being analyzed. Each category offers distinct advantages and is suited to different types of anomalies and data environments, ensuring robust anomaly detection across various applications and industries [5], [6], [7].

This paper investigates anomaly detection methods, focusing on the comparison between a straightforward statistical technique, Z-score, and a more sophisticated approach using autoencoders. The Z-score method relies on statistical measures such as mean and standard deviation to identify anomalies based on deviations from the normal distribution. In contrast, autoencoders are employed to learn and reconstruct data patterns, detecting anomalies through significant reconstruction errors.

Section I introduces the concept of anomalies and provides an overview of commonly employed methods for anomaly detection. Section II details the algorithms and methodologies utilized in this study, encompassing both traditional statistical approaches and advanced machine learning techniques such as autoencoders. Section III presents the outcomes and findings of the experimental evaluation, illustrating the comparative performance and effectiveness of the discussed methods in detecting anomalies for the randomly generated big datasets.

II. Methodology

A. Z-score statistical technique

The Z score, also known as the standard score, is a crucial statistical concept used to determine how far a data point deviates from the mean of a dataset and in which direction. It provides a standardized measure that indicates whether a data value is above or below the mean, as well as the magnitude of this deviation in terms of standard deviations. Specifically, a Z score quantifies the distance between a data point and the mean in units of standard deviation, offering insights into

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

the relative position and significance of the data point within the distribution. This metric is widely utilized across various disciplines to assess outliers, evaluate statistical significance, and normalize data for comparative analysis [8], [9].

Mathematically, Z score can be represented as

$$Zscore = \frac{(x - \mu)}{\sigma} \quad (1)$$

where, x is the data point, μ is the mean and σ is the standard deviation

The following algorithm provides a straightforward method to compute Z scores of the randomly generated big dataset for anomaly detection.

Algorithm 1 : Z score algorithm

Input:
$DataVector : X = [x_1, x_2, ..., x_n]$,
$Mean : \mu$,
$Standard\ Deviation : \sigma$
Output:
$Zscorevector : Z = [z_1, z_2, ..., z_n]$

Step1. Generate synthetic normal data representing normal behavior and synthetic anomaly data simulating anomalous behavior.
Step2. Combine normal data and anomaly data into a single dataset (data).
Step3. Standardize the data to transform data to have zero mean and unit variance.
Step4. Apply PCA for dimensionality reduction to visualize high-dimensional data in 2D.
Step5. Compute the Z-score for each data point after PCA transformation using

$$Zscore = \frac{(x-\mu)}{\sigma}$$

Step6. Define a threshold (threshold) for anomaly detection based on Z-score.
Step7. Identify data points with Z-score greater than the threshold as anomalies.
Step8. Plot data points where their Z-score exceeds the threshold, highlighting potential anomalies.

B. Autoencoders

Autoencoders represent a specialized category of algorithms adept at learning efficient representations of input data without relying on labeled examples. These artificial neural networks are tailored for unsupervised learning, emphasizing the ability to compress and accurately represent input data [10], [11]. Central to their operation is a dual-component structure comprising an encoder and a decoder. The encoder transforms input data into a lower-dimensional representation known as the ”latent space” or ”encoding”, while the decoder reconstructs the original input from this encoded representation. This process allows the network to discern meaningful patterns within the data, facilitating the extraction of essential features crucial for various applications in fields such as image processing and anomaly detection [12], [13].

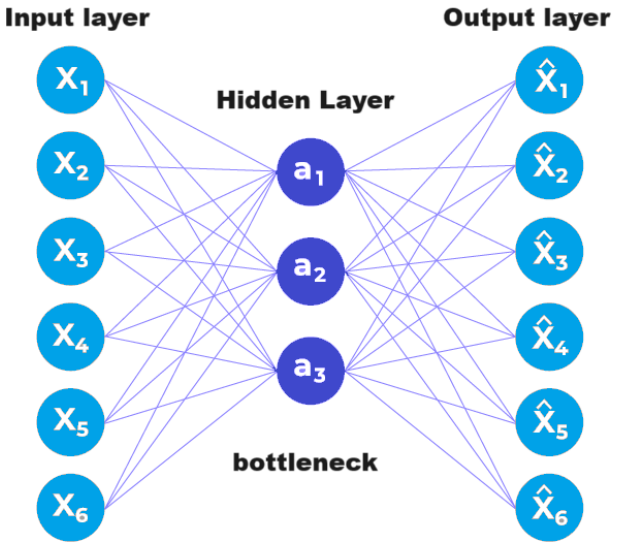


Fig. 1. Autoencoder general architecture

An autoencoder is structured with three main parts: an encoder, decoder, and bottleneck layer as in fig. 1. The encoder starts with an input layer that takes in raw data [14], [15]. As the data moves through hidden layers, these layers progressively reduce its dimensionality while capturing important features and patterns. The final hidden layer, known as the bottleneck layer or latent space, significantly compresses the data to form a condensed representation [16], [17], [18].

On the other hand, the decoder takes this compressed representation from the bottleneck layer and reconstructs it back to the original dimensionality of the input data. Like the encoder, the decoder also has hidden layers that gradually increase the dimensionality, aiming to produce a reconstructed output that closely resembles the initial input. The output layer then generates this reconstructed output.

During training, autoencoders use a loss function, such as mean squared error (MSE) for continuous data or binary cross-entropy for binary data, to measure the difference between the input and the reconstructed output. This loss function guides the network to minimize discrepancies during reconstruction, pushing it to capture the most important features of the input data in the bottleneck layer.

After training, typically only the encoder part of the autoencoder remains in use. It’s employed to encode similar types of data that were used during training, allowing it to effectively distill relevant features from the input data into a compact representation within the latent space [19], [20], [21], [22].

The following algorithm outlines the structure of an autoencoder applied to a randomly generated large dataset for anomaly detection, focusing on its encoding and decoding processes to identify deviations from normal data patterns.

Algorithm 2 : Autoencoder algorithm

Input:

$DataVector : X = [x_1, x_2, ..., x_n]$

Output:

$MeanSquarederror = \mu_e$

Step1. Create two types of data : normal data points (1,000 samples) and anomalous data points (50 samples). These are randomly generated based on normal distributions, but the anomalous data has a different mean.

Step2. Standardize the combined dataset (normal + anomaly) to have zero mean and unit variance.

Step3. Split the standardized data into training and testing sets (80% for training and 20% for testing).

Step4. Define the autoencoder model with encoder using a dense layer with ReLU activation and decoder using another dense layer with ReLU activation.

Step5. Train with Adam optimizer and mean squared error (MSE) loss function.

Step6. Reconstructs all data points and calculate the Mean Squared Error (MSE) between original and reconstructed data points.

Step7. Classify the data points with MSE above 95th percentile threshold as anomalies.

Step8. Calculate performance metrics such as Precision, Recall, and F1-score.

Step9. Plot the original data points, marking anomalies in red, and normal points in blue on a scatter plot.

III. Results and Discussion

A large-scale dataset is randomly generated to evaluate the anomaly detection methods, focusing on precision, F1 score, and recall as key evaluation metrics. Precision quantifies the accuracy of anomaly identification by measuring the ratio of correctly detected anomalies to all identified anomalies. F1 score harmonizes precision and recall, offering a balanced assessment of a method's ability to precisely identify anomalies while capturing all true positives. Recall gauges the method's effectiveness in correctly identifying anomalies among all actual anomalies present in the dataset. These metrics collectively provide a robust framework for assessing the performance and reliability of anomaly detection methods when applied to large, randomly generated datasets, essential for ensuring their practical applicability and effectiveness in real-world scenarios.

A. Case A

The Z score based method implementation integrates PCA for dimensionality reduction and Z-score calculation to detect anomalies in a dataset, which combines synthetic normal and synthetic anomaly data. It standardizes the data to ensure zero mean and unit variance, facilitating effective anomaly detection using a specified threshold. Evaluation metrics precision, recall, and F1-score are computed based on ground truth

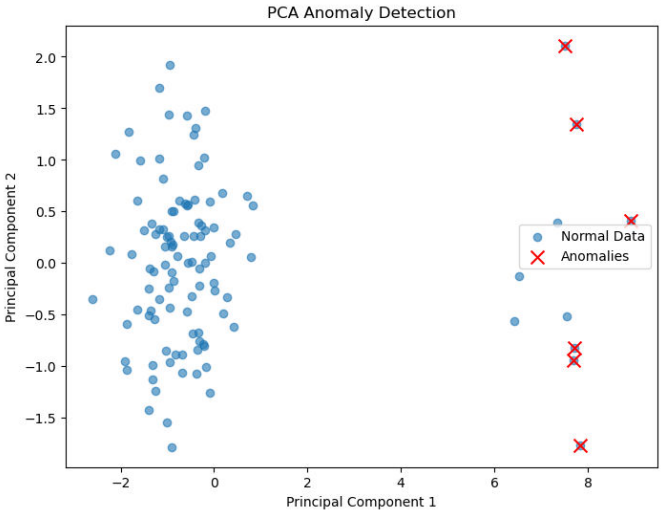


Fig. 2. Anomaly detection using Z score

labels, allowing assessment of the algorithm's performance in correctly identifying anomalies as in fig. 2. Here, precision is 1.00, recall is 0.60 and F1 Score is 0.75.

B. Case B

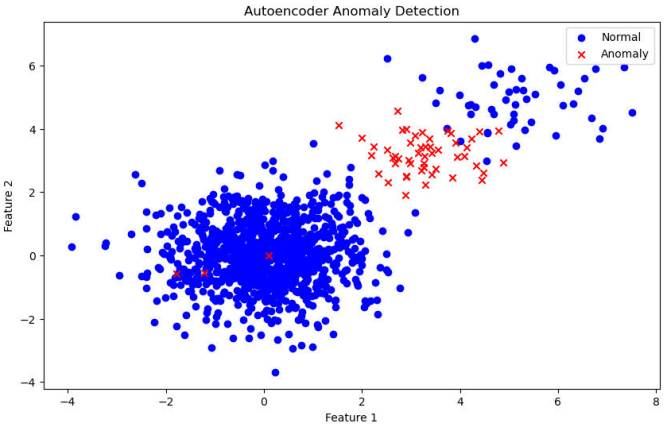


Fig. 3. Scatterplot visualization for anomaly detection using autoencoder

The anomaly detection process using an autoencoder successfully identifies anomalous data points with high precision (0.94), recall (1.00), and an F1-score of 0.97. This indicates that the model accurately detects anomalies while minimizing false positives. The autoencoder, trained on standardized data with an Adam optimizer and MSE loss function, reconstructs data points and calculates reconstruction errors. Anomalies are identified using a threshold set at the 95th percentile of MSE values. The scatter plot visualization in fig. 3 clearly distinguishes anomalous (red) and normal (blue) data points, showcasing the model's effective anomaly detection capability.

Table I compares the performance metrics of two anomaly detection methods: the Z-score based method using PCA and the autoencoder approach. For the Z-score based method,

TABLE I
PERFORMANCE COMPARISON

	Performance parameters		
	precision	*recall*	*F1-score*
Z-score statistical technique	1.00	0.60	0.75
Autoencoders	0.94	1.00	0.97

precision is reported as 1.00, indicating that all identified anomalies were correct, while recall is 0.60, suggesting that 60% of actual anomalies were detected. The F1-score, which balances precision and recall, is 0.75, reflecting the overall effectiveness of this method.

In contrast, the autoencoder method achieves a precision of 0.94, indicating a high proportion of correctly identified anomalies with minimal false positives. The recall is 1.00, indicating that all actual anomalies were detected. Consequently, the F1-score for the autoencoder method is 0.97, signifying superior performance compared to the Z-score based approach. This comparison highlights that the autoencoder, leveraging its ability to reconstruct data and detect anomalies based on reconstruction errors, achieves more robust and accurate anomaly detection results across the evaluated metrics.

IV. CONCLUSIONS

This work demonstrates that autoencoders significantly outperform the traditional Z-score statistical technique in detecting anomalies within large datasets. With precision, recall, and F1-score metrics of 0.94, 1.00, and 0.97 respectively, autoencoders prove to be more effective at accurately identifying anomalies. This superior performance underscores their suitability for enhancing operational reliability and minimizing disruptions in complex and high-volume data environments. Industries can use autoencoders to detect anomalies effectively, improving decision-making and minimizing costly downtime.

REFERENCES

[1] Liso, Adriano, Angelo Cardellicchio, Cosimo Patruno, Massimiliano Nitti, Pierfrancesco Ardino, Ettore Stella, and Vito Ren. "A review of deep learning based anomaly detection strategies in Industry 4.0 focused on application fields, sensing equipment and algorithms." IEEE Access (2024).
[2] Nawaz, Ali, Shehroz S. Khan, and Amir Ahmad. "Ensemble of Autoencoders for Anomaly Detection in Biomedical Data: A Narrative Review." IEEE Access (2024).
[3] DSouza, Divya Jennifer, and K. R. Uday Kumar Reddy. "Anomaly detection for big data using efficient techniques: A review." In International Conference on Artificial Intelligence and Data Engineering, pp. 1067-1080. Singapore: Springer Nature Singapore, 2019.
[4] Rana, Annie Ibrahim, Giovani Estrada, Marc Sol, and Victor Munts. "Anomaly detection guidelines for data streams in big data." In 2016 3rd International Conference on Soft Computing and Machine Intelligence (ISCMI), pp. 94-98. IEEE, 2016.
[5] Alghushairy, Omar, Raed Alsini, Terence Soule, and Xiaogang Ma. "A review of local outlier factor algorithms for outlier detection in big data streams." Big Data and Cognitive Computing 5, no. 1 (2020): 1.
[6] Thudumu, Srikanth, Philip Branch, Jiong Jin, and Jugdutt Singh. "A comprehensive survey of anomaly detection techniques for high dimensional big data." Journal of Big Data 7 (2020): 1-30.
[7] Palakurti, Naga Ramesh. "Challenges and future directions in anomaly detection." In Practical Applications of Data Processing, Algorithms, and Modeling, pp. 269-284. IGI Global, 2024.
[8] Chikodili, Nwodo Benita, Mohammed D. Abdulmalik, Opeyemi A. Abisoye, and Sulaimon A. Bashir. "Outlier detection in multivariate time series data using a fusion of K-medoid, standardized euclidean distance and Z-score." In International Conference on Information and Communication Technology and Applications, pp. 259-271. Cham: Springer International Publishing, 2020.
[9] Habeeb, Riyaz Ahamed Ariyaluran, Fariza Nasaruddin, Abdullah Gani, Ibrahim Abaker Targio Hashem, Ejaz Ahmed, and Muhammad Imran. "Real-time big data processing for anomaly detection: A survey." International Journal of Information Management 45 (2019): 289-307.
[10] Nawaz, Ali, Shehroz S. Khan, and Amir Ahmad. "Ensemble of Autoencoders for Anomaly Detection in Biomedical Data: A Narrative Review." IEEE Access (2024).
[11] Zhou, Chong, and Randy C. Paffenroth. "Anomaly detection with robust deep autoencoders." In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 665-674. 2017.
[12] Borghesi, Andrea, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. "A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems." Engineering Applications of Artificial Intelligence 85 (2019): 634-644.
[13] Ahmad, Sabtain, Kevin Styp-Rekowski, Sasho Nedelkoski, and Odej Kao. "Autoencoder-based condition monitoring and anomaly detection method for rotating machines." In 2020 IEEE International Conference on Big Data (Big Data), pp. 4093-4102. IEEE, 2020.
[14] Maleki, Sepehr, Sasan Maleki, and Nicholas R. Jennings. "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering." Applied Soft Computing 108 (2021): 107443.
[15] Bian, Yihan, and Xinchen Tang. "Abnormal detection in big data video with an improved autoencoder." Computational Intelligence and Neuroscience 2021, no. 1 (2021): 9861533.
[16] Singh, Richa, Nidhi Srivastava, and Ashwani Kumar. "Network Anomaly Detection Using Autoencoder on Various Datasets: A Comprehensive Review." Recent Patents on Engineering 18, no. 9 (2024): 63-77.
[17] "Autoencoder-based Anomaly Detection System for Online Data Quality Monitoring of the CMS Electromagnetic Calorimeter." Computing and Software for Big Science 8, no. 1 (2024): 11.
[18] Wang, Kai, Caoyin Yan, Yanfang Mo, Yalin Wang, Xiaofeng Yuan, and Chenliang Liu. "Anomaly detection using large-scale multimode industrial data: An integration method of nonstationary kernel and autoencoder." Engineering Applications of Artificial Intelligence 131 (2024): 107839.
[19] Shrestha, Rakesh, Mohammadreza Mohammadi, Sima Sinaei, Alberto Salcines, David Pampliega, Raul Clemente, Ana Lourdes Sanz, Ehsan Nowroozi, and Anders Lindgren. "Anomaly detection based on lstm and autoencoders using federated learning in smart electric grid." Journal of Parallel and Distributed Computing (2024): 104951.
[20] Gogineni, Vinay Chakravarthi, Katinka Mller, Milica Orlandi, and Stefan Werner. "Lightweight Autonomous Autoencoders for Timely Hyperspectral Anomaly Detection." IEEE Geoscience and Remote Sensing Letters (2024).
[21] Jia, Watson, Raj Mani Shukla, and Shamik Sengupta. "Anomaly detection using supervised learning and multiple statistical methods." In 2019 18th IEEE international conference on machine learning and applications (ICMLA), pp. 1291-1297. IEEE, 2019.
[22] Maleki, Sepehr, Sasan Maleki, and Nicholas R. Jennings. "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering." Applied Soft Computing 108 (2021): 107443.

Grapevine Leaf Disease Classification using Deep Convolutional Neural Networks

Md Al-Imran
Dept. of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
al.imran@ewubd.edu

Sajid Faysal Fahim
Dept. of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
sajidfaysalfahim@gmail.com

Sanjida Simla
Dept. of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
2019-2-50-007@std.ewubd.edu

Fatin Hasnat Shakib
Dept. of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
shakibbenyaamin@gmail.com

Tokiuddin Ahmed
Dept. of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
t.ahmedchayan@gmail.com

Sarwar Jahan
Department of Computer Science and Engineering East West University,
Dhaka-1212, Bangladesh
sjahan@ewubd.edu

***Abstract*— The existence of numerous diseases that afflict the fragile and lush leaves of grapevines, bringing with them their destructive impact, presents a looming and formidable danger to the meticulous and intricate art of cultivating grapes. This peril not only puts at risk the bountiful harvest but also endangers the unmatched brilliance and refinement of the resulting crop. In the present study, we employ cutting-edge Deep Convolutional Neural Networks (DCNNs) to present a fresh and innovative approach to identifying disorders that affect grapevine leaves. Our methodology involves the fine-tuning of pre-existing DCNN models, utilizing a tailor-made dataset composed of a wide array of images showcasing grapevine leaves affected by various diseases. This approach facilitates precise classification and reliable generalization. The far-reaching consequences of this investigation extend far beyond the mere advancement of grapevine disease detection. It lays the groundwork for a comprehensive system that might dramatically revolutionize the method by which we monitor illnesses in vineyards. This has the opportunity to not only transform the world of viticulture but also open the door to more environmentally friendly approaches that will eventually lead us to a more affluent and ecologically conscious future.**

Keywords*— *Deep Learning, Convolution Neural Network (CNN), Feature Extraction, Image Processing, Transfer Learning.

I. INTRODUCTION

Grapes are a global agricultural mainstay, contributing significantly to the manufacture of wines, juices, and raisins. However, several leaf diseases pose an ever-increasing hazard to the grapevine, threatening both productivity and quality. To avoid substantial crop damage and economic losses, it is vital to diagnose and categorize these infections as soon as feasible. Traditional disease detection methods in viticulture frequently rely on manual inspection, which may be time-consuming and subjective [1]

.

A. *Contribution*

- classify grapevine leaf diseases using DCNNs.
- We aim to construct a robust and reliable classification system by using pre-trained models and the development of a custom CNN model as feature extractors and fine-tuning them on a curated dataset including various photos of sick grapevine leaves.
- This technique not only overcomes the issues associated with illness pattern variability, but it also capitalizes on the amount of information inherent in pre-existing models, allowing the network to detect minor traits indicative of certain diseases.
- Various deep learning and other technologies are utilized to classify leaf diseases.
- A research gap has been identified, and additional study has been suggested.

This noble research undertaking aspires to unlock the limitless possibilities offered by Deep Convolutional Neural Networks (DCNNs), to proficiently classify different types of grapevine leaf disorders. To accomplish this, we will employ pre-trained models as tools to extract distinctive features and then refine them using a meticulously curated dataset that encompasses a wide array of images depicting diseased grapevine leaves. The ultimate objective of this undertaking is to develop a classification system that is not only resilient but also exceptionally precise. By adopting this methodology, we can address the inherent challenges associated with the variability of disease patterns while simultaneously leveraging the extensive pool of knowledge embedded within pre-existing models. Consequently, the network becomes empowered to discern even the most subtle characteristics that are indicative of specific diseases and contribute to the overall effectiveness of the classification system.



(a) **(b)** **(c)** **(d)**

Fig.1. Sample image of grapevine leaves: (a) black rot, (b) esca, (c) healthy, and (d) leaf blight.

Below is a summary of the paper's structure: Section 2 includes work that has already been completed in the

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

discovery of diseases. Section 3 discusses an overview of deep learning. The steps for diagnosing a grapevine Leaf disease are described in Section 4. Section 5 addresses the results and discussion of the current research, so to achieve effective grapevine leaf disease identification systems, these issues must be resolved. Observation and conclusion are described in Section 6, respectively.

II. LITERATURE REVIEW

Durmus et al. [2] conducted an investigation aimed at identifying diseases in tomato leaves. The authors utilized the advanced deep learning frameworks of AlexNet and SqueezeNet for this purpose. The images they analyzed were sourced from the extensive Plant-Village database, which encompassed a diverse range of 14 crops. The focus of their work was solely on tomato leaves, which were classified into 10 different categories. Using AlexNet, they attained a phenomenal accuracy rate of nearly 95%. However, when it came to considerations such as model size and inference time, the SqueezeNet structure outperformed its counterpart.

Johannes et al. [3] undertook a captivating endeavor, delving into the realm of more than 3000 mesmerizing images, each showcasing one of the 38 enchanting varieties of wheat plants in their untamed habitat. Their noble quest was to unveil the secrets of three distinct European endemic wheat diseases. In their quest, they unveiled a remarkable fusion of a timeless machine learning model and an awe-inspiring statistical inference method.

Jain et al. created a cloud-based system that uses predictive learning and neural networks based on convolution to address agricultural health in distant areas of India [4]. They performed real-time categorization of sick plant pictures and compared techniques to reduce misclassification.

Picon et al. [5] refreshed the robotized multi-illness recognition method shown by Johannes et al. for field securing settings. They utilized DCNN to identify beginning phase illnesses and concurrent infections, effectively approving similar three illnesses over 8000 pictures and tried under genuine field conditions[6].

III. OVERVIEW OF DEEP LEARNING

Deep Learning (DL) is a subset of machine learning that includes using brain networks with different layers, otherwise called Deep brain organizations.[7]. These organizations are intended to learn and address information through the iterative course of preparing a lot of named data. Deep Learning has acquired noticeable quality for its striking skill to display and break down complex examples, empowering applications across different areas, including computer vision, natural language processing, speech recognition, and more[8]. Deep learning does not require the extra pre-processing step of machine learning named as feature extraction which provides it an upper hand over machine learning [9].

A. CNN Architecture

CNN structures follow the fundamental guideline of applying progressive convolution layers alongside pooling layers and completely associated layers to the contribution to down example the dimensions without losing the significant highlights caught by the highlights map and furnishing the necessary result with the most extreme precision conceivable[10]. Convolution is applied to the output of the fully connected layer and predicts the picture class using the previously extracted features[11].

- ***Convolution Layer:*** The Convolutional Layer is the initial layer that focuses on extracting characteristics from the input pictures[12]. This layer performs the convolution mathematical procedure between each input picture and a series of convolution filters of varying sizes. Swiping the filter over the input picture generates the dot result between the filter and the portions of the image according to the filter's size [13].

- ***Pooling Layer:*** Pooling in convolutional neural systems may be achieved in two ways. Maximum and average pooling. In Max Pooling, which is the most popular of the two, we scan the highest possible value for each region of the image. Average pooling computes the mean of the components of a picture inside a predetermined sized zone. The Pooling Layer connects the Convolutional neural network Layer with the Fully Connected Layer.

- ***Flattening:*** It flattens the information into a set so that CNN can read it. It is often delivered ahead of the commencement of the completely linked layer.

- ***Fully Connected:*** We flatten the final result of the previous layer of convolution in this layer and connect every node in the present layer to all the others in the following layer.

- ***Activation:*** Activation functions play a crucial and pivotal role in the intricate dance of neural networks. These functions hold the power to decide which delicate strands of information should gracefully waltz forward in the network's journey, while gracefully dismissing others at the end of their performance. In this enchanting process, they sprinkle a touch of nonlinearity, adding a dash of magic to the network's tapestry.

  ReLU: The rectified linear unit activation function removes values that are negative from the feature maps before producing an attribute map with only non-negative values.

  $$\mathrm{ReLU} = \max(0,x)$$

  Softmax: By supplying probability distributions from the input vectors, it aids in forecasting the correct maximum from multiple classes of data. The values vary between 0 to 1 [14].

IV. STEPS FOR DIAGNOSING A LEAF DISEASE

CNN calculations investigate a picture and concentrate on its highlights. Convolutional neural networks are deep learning calculations that can interact with enormous datasets containing a large number of boundaries, demonstrated on 2D pictures, and interface the subsequent portrayals to the comparing yields. A CNN is a regulated multi-facet network that can progressively advance new highlights from datasets.

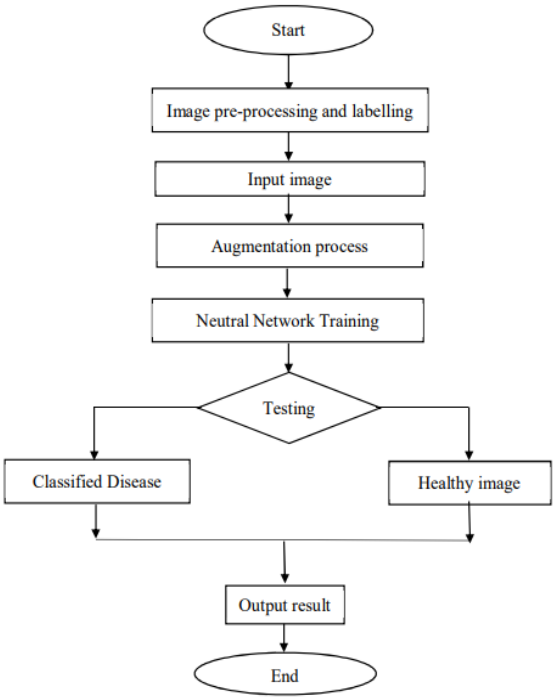


Fig. 2: Flow Chart

- ***Dataset Description:*** This dataset consists of 9027 images of diseased and healthy plant leaves of the Plant Village dataset, which were classified into 4 classes to train a deep convolutional neural network that can identify the grapevine leaf diseases.
- ***Data Preprocessing:*** The data will be divided into three distinct classifications: Preparing, Validation, and Testing. 70% of the dataset is for training, 20% for Validation, and 10% for Testing. The preparation information will be utilized to prepare the deep learning CNN model, and its boundaries will be calibrated with the approval information. At long last, the presentation of the data will be assessed utilizing the test information.
- ***Data Augmentation:*** A strategy for boosting the number of photographs in a database is data augmentation. To diversify our dataset, we apply multiple techniques to picture datasets, such as changing, turning, zooming, and rotating. Overfitting may be mitigated during the training stage by enriching the dataset and introducing deformation to the pictures. The image data generator class in Keras supports in-place or on-the-fly data augmentation. With this form of data augmentation, we can ensure that our trained network sees fresh changes at each epoch. It enables us to achieve great outcomes while working with a smaller dataset [15]

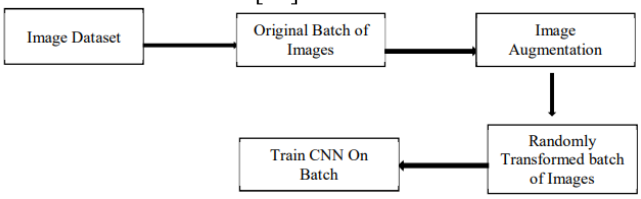


Fig. 3: Data Augmentation

A remarkable and ingenious convolution instrument that possesses the remarkable ability to meticulously and methodically disentangle and classify the diverse and multifarious attributes and characteristics that are inherent within images, thereby rendering them suitable and amenable for thorough scrutiny and examination in what is commonly referred to as the process of feature extraction, a technique that entails the identification and segregation of the distinctive and noteworthy aspects and elements of the said images.

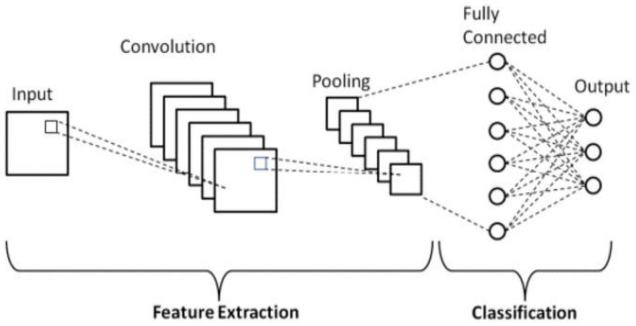


Fig. 4: CNN Architecture

The Sequential Model API enables the creation and addition of sequential classes and model layers in deep learning models. The input to a convolutional neural network is an (n x m x 3) matrix for colored images, with the number 3 representing the RGB components of each pixel. The model is made up of numerous 2D convolutional layers with 64 filters and a rectified linear unit with various filters and activation functions. This is followed by sequential normalization, pooling, and fully connected 32 filters, followed by a layer before applying the softmax activation function to probability values.

- ***Training the Model:*** During the subsequent stage, the model is assembled and trained on a specially designed dataset. To compile the model successfully, specific parameters need to be defined, such as the optimizer. The optimizer, like Adam, modifies the weights and learning rate of a neural network to reduce losses efficiently. Models may be trained quicker and more successfully by using optimizers. We select the Adam optimization technique for our multi-class classification problem because it consistently produces smoother results. The loss function is another critical parameter, and

we choose the "sparse categorical cross entropy" loss function for our integer targets [16]. When fitting the model, we consider additional parameters like the batch size is 32 and the number of epochs. For our training process, we train the model over 100 epochs to enhance accuracy and reduce loss [17].

V. RESULTS AND DISCUSSIONS

The findings of using deep convolutional neural networks to classify grapevine leaf diseases reveal not only remarkable accuracy but also outstanding efficiency over a broad range of designs. To visually capture the highest level of accuracy achieved by the model during both the training and validation stages, we meticulously create an illustrative graph that vividly depicts the maximum accuracy reached while simultaneously minimizing the loss function. This graph is a visual depiction of the model's knowledge and serves as evidence of the efficacy of the approaches used.

A. EfficientNetB0 Accuracy (99.94%):

EfficientNetB0, a renowned and highly acclaimed neural network architecture celebrated for its unparalleled efficiency and remarkable performance, has showcased its true prowess by attaining a remarkable accuracy rate of an astounding 99.94%. This exceptional feat serves as a testament to its remarkable ability to discern and capture even the most intricate and minute patterns and features that hold immense significance in the realm of grapevine leaf diseases. The sheer magnitude of this remarkable accuracy rate not only underscores the effectiveness of the model but also serves as a resounding affirmation that it has triumphantly surmounted the challenges of generalizing its acquired knowledge from pre-training to meticulously and flawlessly classify an extensive range of diverse instances of diseases present within the hallowed realm of grapevine leaves.

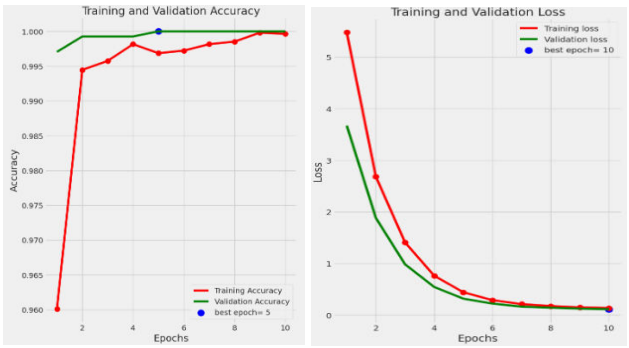


Fig. 5: training v/s validation loss and accuracy

B. Xception Accuracy (99.88%):

Xception, a marvel of modern deep learning architecture, has proven its worth through the utilization of depth-wise separable convolutions, a technique that has yielded remarkable outcomes. The network's astounding accuracy rate of 99.88% serves as a testament to its prowess in comprehending intricate spatial hierarchies and discerning complex patterns. The network's exceptional ability to capture the essence of the grapevine leaf disease dataset is evident in its superior classification accuracy. It is worth noting that the slightly lower accuracy achieved compared to EfficientNetB0 can be attributed to the distinctive architectural nuances and the unique characteristics of the dataset about grapevine leaf diseases.

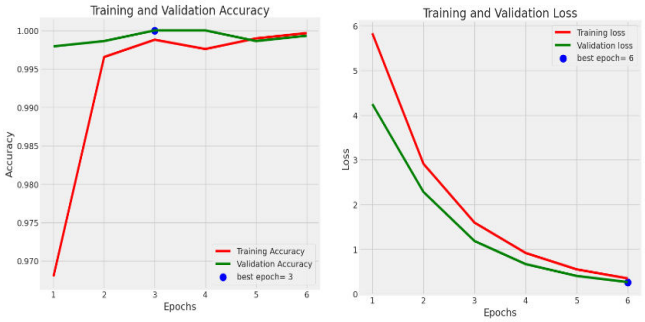


Fig. 6: training v/s validation loss and accuracy of Xception

C. MobileNetV3 Accuracy (99.45%):

MobileNetV3, which has been specifically engineered to optimize efficiency and cater to the needs of mobile applications, has showcased an impressive level of accuracy, achieving a commendable 99.45% accuracy rate. While this accuracy slightly falls short when compared to the other two architectural models, MobileNetV3 continues to demonstrate its remarkable effectiveness in the realm of grapevine leaf disease classification. With its ability to effectively classify and identify various diseases affecting grapevine leaves, MobileNetV3 emerges as a highly practical and viable option for deployment in environments that are constrained by limited resources.

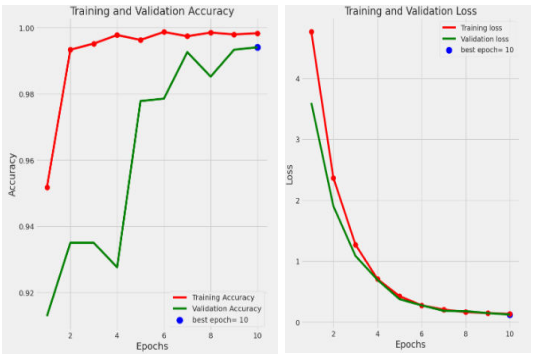


Fig. 7: training v/s validation loss and accuracy of MobileNetV3

D. Custom CNN Accuracy (99.48%):

A revolutionary technique is being used to categorize photos using a CNN, which eliminates the requirement for pre-trained models. While pre-trained models can distinguish between numerous classes without individual training, their intricate architectures may pose challenges for beginners. To simplify the process of constructing custom CNN, Keras was utilized in the development of this project [18].

The Sequential model is utilized. A sequential model API is a method for constructing deep learning models with sequential classes and model layers. In a convolutional neural network, the input for colored images is represented as (n x m x 3). The model starts with a 2D convolutional layer using 32 filters and Rectified Linear Unit activation. Batch normalization and maximum pooling are then applied in subsequent layers. Two blocks of a 2D Convolutional layer with 64 filters and Rectified Linear Unit activation are added, followed by a pooling layer. Finally, a Convolutional layer with 32 filters, Rectified Linear Unit activation, and pooling are included. The output is flattened for fully connected layers. An additional 512 dense layers with a dropout of 0.2 are added. The softmax activation function is used for converting outputs into probability values.

The accuracy of the custom CNN, which is 99.48%, shows that it can compete with advanced pre-trained models. The custom architecture, made for grapevine leaf diseases, proves that a tailored approach can achieve similar results to highly optimized pre-trained models.

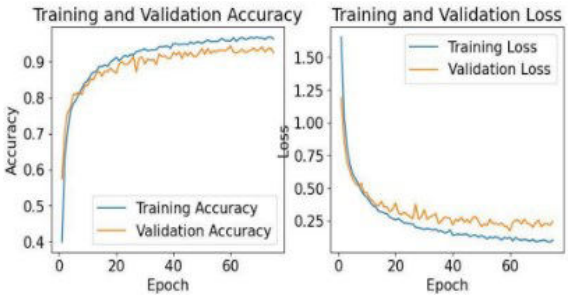


Fig. 8: training v/s validation loss and accuracy of Custom CNN

E. Confusion Matrix

A confusion matrix, a beautiful and intricate creation, serves as a grand table that elegantly showcases the magnificent performance of a classification model. It is a visual masterpiece that transports us to a realm where correct and incorrect predictions dance harmoniously with one another, expressing their true essence for each class.

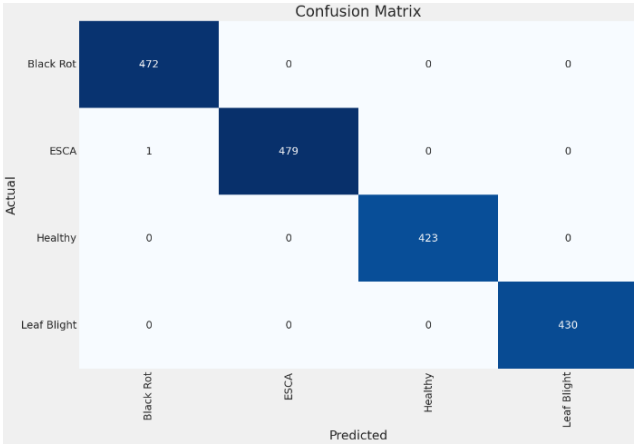


Fig. 9: confusion matrix of EfficientNetB0

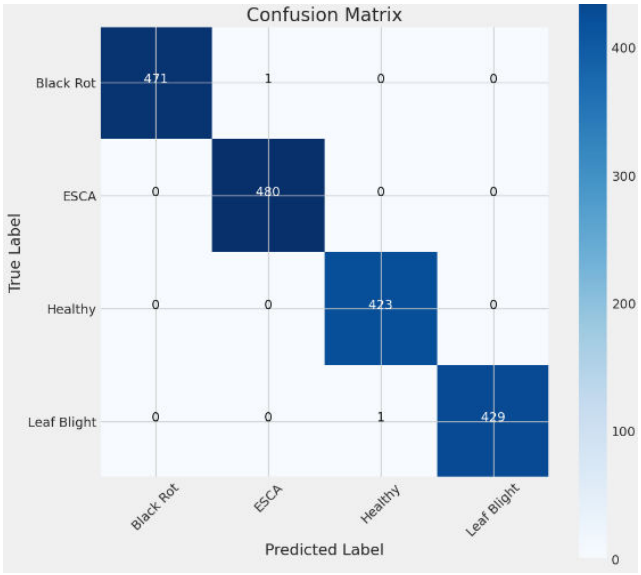


Fig. 10: confusion matrix of Xception

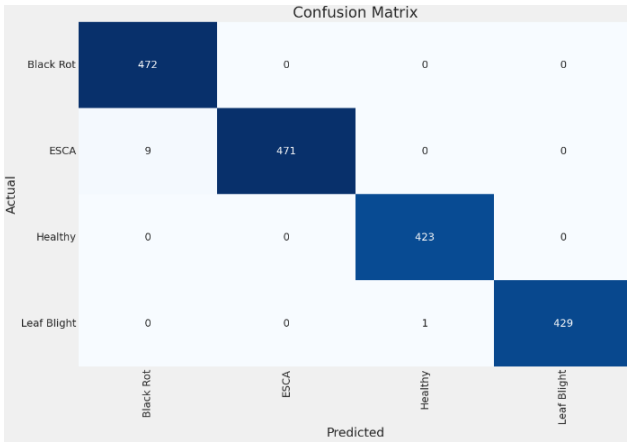


Fig. 11: confusion matrix of MobileNetV3

In the mesmerizing world of grapevine leaf disease classification, where the powerful forces of EfficientNetB0, Xception, and MobileNetV3 intertwine, the creation of a confusion matrix becomes an absolute necessity. It is a wondrous spectacle that unfolds before our eyes as each model gracefully unfolds its unique rendition of the confusion

matrix, a masterpiece that captures the very essence of their predictions. The impressive accuracies achieved by all four architectures highlight the power of transfer learning in classifying grapevine leaf diseases.

Table 1: Proposed Method Result

Implemented method	Disease name	No.of image	Dataset	Accuracy
EfficientNetB0	black rot, esca, leaf blight	9027	plant village	99.94%
Xception	black rot, esca, leaf blight	9027	plant village	99.88%
MobileNetV3	black rot, esca, leaf blight	9027	plant village	99.45%
Custom CNN	black rot, esca, leaf blight	9027	plant village	99.48%

EfficientNetB0's exceptional accuracy is attributed to its unique scaling method and compound scaling coefficients, which effectively extract essential features. Xception, although slightly less accurate, excels at capturing intricate details due to its architectural choices. MobileNetV3, with its commendable accuracy, is particularly suitable for resource-constrained applications. When compared to pre-trained models, a custom CNN that is particularly customized to the unique features of the dataset can obtain comparable performance. Overall, our findings verify the efficacy of deep convolutional neural networks in grapevine leaf disease classification, opening the door to their implementation in accurate agricultural systems.

VI. CONCLUSION

The capabilities of convolutional neural networks have advanced dramatically in recent years. The latest iteration of these networks has displayed promising outcomes in the domain of image recognition. This project explored a unique method of automatically categorizing and identifying plant diseases from leaf images by employing sophisticated deep-learning techniques. With an astonishing accuracy of 99.94%, 99.88%, 99.45%, and 99.48%, the custom CNN model developed was able to differentiate healthy leaves from three visually identifiable diseases. Given this outstanding level of execution, it is clear that convolutional neural networks are uniquely suited for illness identification and detection.

REFERENCES

[1] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Ste- fanatic, "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection", Symmetry 2019, 11,939; doi:10.3390/sym11070939.

[2] https://towardsdatascience.com/what-is-deep-learning-and-how-does- it-work-2ce44bb692ac

[3] MK Gurucharan. "Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | UpGrad Blog." UpGrad Blog, 7 Dec. 2020.

[4] Adrian Rosebrock. "Keras ImageDataGenerator and Data Augmentation PyImageSearch." PyImageSearch, 8 July 2019.

[5] Michael A. Nielsen. "Neural Networks and Deep Learning." Neural Networks and Deep Learning.

[6] Arsenovic Marko, Karanovic Mirjana, Sladojevic S. "Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection." Symmetry. 2019; 11(7):939.

[7] B. Sandika, S. Avil, S. Sanat and P. Srinivasu. "Random forest-based classification of diseases in grapes from images captured in uncontrolled environments," 2016 IEEE 13th International Conference on Signal Processing (ICSP), 2016, pp. 1775-1780, doi: 10.1109/ICSP.2016.7878133.

[8] H. Durmu, O. Güne, and M. Krc, "Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning," in 6th International Conference on Agro-Geoinformatics, 2017.doi: DOI:10.1109/AGRO-GEOINFORMATICS.2017.8047016

[9] A. Johannes et al., "Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case," Comput Electron Agric, vol. 138, pp. 200–209, Jun. 2017, doi: 10.1016/j.compag.2017.04.013

[10] L. Jain, M. A. H. Vardhan, M. L. Nishanth, and S. S. Shylaja, "Cloud- based system for supervised classification of plant diseases using convolutional neural networks," in Proceedings - 2017 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2017, Apr. 2018, vol. 2018-January, pp. 63–68. doi: 10.1109/CCEM.2017.22.

[11] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," Comput Electron Agric, vol. 161, pp. 280–290, Jun. 2019, doi: 10.1016/j.compag.2018.04.002.

[12] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," Comput Electron Agric, vol. 154, pp. 18–24, Nov. 2018, doi: 10.1016/j.compag.2018.08.048.

[13] M. Zhang and Q. Meng, "Automatic citrus canker detection from leaf images captured in the field," Pattern Recognit Lett, vol. 32, no. 15, pp. 2036–2046, Nov. 2011, doi: 10.1016/j.patrec.2011.08.003.

[14] S. v Militante, B. D. Gerardo, and N. v Dionisio, "Plant Leaf Detection and Disease Recognition using Deep Learning," in 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, pp. 579–582. doi:10.1109/ECICE47484.2019.8942686.

[15] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D.P. Hughes, "Deep learning for image-based cassava disease detection," Frontiers Plant Sci., vol. 8, p. 1852, Oct. 2017.

[16] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," Comput. Electron. Agricult., vol. 154, pp. 18–24, Nov. 2018. doi:10.1016/j.compag.2018.08.048.

[17] Y. Zhong, M. Zhao, "Research on deep learning in apple leaf disease recognition". Computers and Electronics in Agriculture 168 (2020) 105146, https://doi.org/10.1016/j.compag.2019.105146

[18] Monigari, V. (2021). Plant leaf disease prediction. International Journal for Research in Applied Science and Engineering Technology, 9(VII), 1295–1305. https://doi.org/10.22214/ijraset.2021.36582

turnitin Page 162 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

PCB Surface Defect Detection Using Defect-centered Image Generation and Optimized YOLOv8 Architecture

Thongpun Supong
College of Innovative Technology and Engineering,
Dhurakij Pundit University,
Bangkok, Thailand
Email: 65130504@dpu.ac.th

Thanapat Kangkachit
College of Innovative Technology and Engineering,
Dhurakij Pundit University,
Bangkok, Thailand
Email: thanapat.kan@dpu.ac.th

Duangjai Jitkongchuen
Manpower Development Division
Big Data Institute
(Public Organization)
Bangkok, Thailand
Email: duangjai.ji@bdi.or.th

***Abstract*— Defect detection on printed circuit boards (PCBs) is a critical challenge in the electronics manufacturing industry, as undetected defects can lead to financial losses, product recalls, and compromised reliability. Advances in deep learning techniques have made automated defect detection more feasible and effective than manual inspection or classical image processing techniques. Recent studies have utilized the YOLO architecture as a baseline, incorporating various modules to enhance the detection of small defects in large-size PCB images. However, these approaches still face challenges with the input information loss due to the necessity of reducing the PCB input images to a smaller size. To address this issue, we propose a defect-centered image generation method enhanced through intensive augmentation, enabling the model to learn detailed information about defects and their surrounding environments. Additionally, several modifications on the YOLO architecture are made including convolutional block attention Module (CBAM) and feature pyramid network (FPN), to improve the detection of small defects. Experimental results demonstrate that our method outperforms existing approaches in terms of mAP50 and recall, while maintaining comparable precision. Specifically, our method archives near-perfect mAP50 score for nearly all defect types, with only a few false positives. These findings underscore the model's superior accuracy and effectiveness in defect detection, making it valuable for real-world industrial applications.**

Keywords*— *Printed Circuit Boards (PCBs), Defect Detection, Deep Learning, YOLO Architecture, Defected-Center Image Generation, Convolutional Block Attention Module (CBAM), Feature Pyramid Network (FPN)

I. Introduction

Printed Circuit Boards (PCBs) are fundamental components in electronic devices, critically influencing product quality and reliability. Ensuring high-quality PCB design and production is essential for maintaining the performance and longevity of electronic devices such as smartphones, computers, and communication equipment. Therefore, effective defect detection methods are necessary to uphold the standards of modern electronics.

Manufacturers have adopted advanced technologies like Automated Optical Inspection (AOI) and Automated Vision Inspection (AVI) systems to quickly and accurately detect defects on PCBs, replacing manual inspection. However, the decreasing size and increasing complexity of PCB designs present challenges, leading to higher detection error rates. Minor production or inspection errors can cause significant issues, resulting in increased costs for rectification and process improvements.

Recent advancements in deep learning have revolutionized the field of defect detection on PCBs. Deep learning techniques, particularly those utilizing convolutional neural networks (CNNs), have emerged as powerful tools for enhancing defect detection capabilities. Among these, the YOLO (You Only Look Once) architecture [6] has demonstrated impressive performance in detect objects with high precision. Recent research has further extended the YOLO architecture with additional modules, such as Convolutional Block Attention Module (CBAM) [10] and Feature Pyramid Network (FPN) [7], to improve the detection of small and complex defect patterns. These enhancements allow the model to focus on small defects and aggregate multi-scale features, resulting in significant improvements in both precision and recall. Nevertheless, these approaches continue to encounter challenges related to information loss, primarily due to the need to resize PCB images, which can decrease detection accuracy.

In this research, the deep learning model is trained to learn detailed patterns of small defects from a comprehensive dataset. This dataset comprises enriched, defect-centered images generated through advanced data augmentation techniques. Additionally, several modules, including the Convolutional Block Attention Module (CBAM) and Feature Pyramid Network (FPN), are integrated into the YOLO architecture to enhance the detection of small defects. The structure of this paper is as follows: a review of related works, a description of the methodology, presentation of experimental results, and a conclusion.

II. Related Works

The rapid development of deep learning algorithms has significantly improved defect detection. Recently, numerous studies [9] have focused on enhancing PCB defect inspection systems, emphasizing improvements in accuracy, precision, recall, and real-time performance as follows.

Du [1] proposed a method for detecting surface defects on PCBs using a dataset containing 693 defect images, augmented with 507 rotated images. The data was divided into training, validation, and test sets in a ratio of 960:120:120. This research utilized an enhanced YOLOv5 model with MBConv Modules, CBAM Attention, BiFPN, SIoU Loss Function, and Depth-Wise Convolutions to improve accuracy and computational efficiency, demonstrating significant improvements in mean accuracy and recall.

Jiang [2] developed a defect detection method using a hybrid model called RAR-SSD. The dataset comprised five types of defect images from the Hikvision MV-CE200-11UC

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

camera, with over 1500 samples per type. Input images were resized to 300x300 pixels and augmented through flipping, reflecting, shifting, and blurring. The data was split into training, validation, and test sets in a 9:1 ratio. This research emphasized accuracy and detection efficiency by utilizing the RFB-s module, SENet, CANet, and a Feature Fusion Module.

Zhang [3] presented a lightweight real-time defect detection system based on an improved YOLOv5s model, incorporating channel pruning to reduce computational load while maintaining accuracy. Human-computer interaction was facilitated through a Raspberry Pi for data transmission and PyQT5 for interface development. The experimental data from the Ku-Market-PCB dataset, split into training and validation sets in a 960:120 ratio, showed that the pruned model achieved sufficient accuracy for real-time applications requiring speed.

Yuan [4] focused on detecting surface defects using 693 images from the HRIPCB dataset from Peking University, split into training and test sets in a 9:1 ratio. An enhanced YOLOv5 model with a HorNet Backbone, MCBAM, CARAFE, and an Optimized Detection Head (DH) was used. The study showed significant improvements in accuracy and efficiency for detecting small PCB defects, making the YOLO-HMC model suitable for real-time industrial applications.

Zhou [5] aimed at detecting small defects on PCBs using a dataset of 693 PCB images with 2,953 defect locations, divided into training, validation, and test sets in a 2096:294:563 ratio. This research utilized an improved YOLOv7 model called TDD-YOLO, incorporating a compression training strategy and a Backbone Network with four Multi-Scale Feature Extraction Layers (ME). TDD-YOLO showed significant improvements in accuracy and speed, particularly with datasets compressed at 0.4 and 0.8 ratios, reducing training time and resources.

III. Methodology

Due to the fact that the defects are small objects in the large PCB board, our approach for training model is to enable it to learn the detailed characteristics of the defects as comprehensively as possible, as illustrated in Fig. 1. To achieve this, we first generate a new training dataset focusing on a single defect per image. By fixing the defect in the center of image, we then crop the entire image to a size of 640x640 pixels. However, training the model with centered detects might reduce its ability to detect defects in other positions on the PCB. To overcome this limitation, various data augmentation techniques are utilized to diversify the defect characteristics learned by the model. This diversification may lead to defect imbalance. Therefore, a Focal Loss layer is incorporated to mitigate issues related to this imbalance, ensuring more robust defect detection. Additionally, a CBAM layer is added to address important feature extraction, and a Feature Pyramid Network (FPN) is employed to improve multi-scale defect detection. For performance evaluation, the original-size defect images are utilized or they are slightly adjusted to fit the stride-32 condition.

A. Data collection

This research utilizes the PKU-Market-PCB [10] dataset, comprising 693 images of PCBs with an average size of 2240x2016 pixels, developed by researchers from Peking University, China, in 2019. The dataset includes 10 different PCB models with varying sizes and circuit patterns as depicted in Fig. 2. The dataset is categorized by defect types, with labels and bounding boxes assigned to each defect on the PCB images.

The defect types include six categories: missing hole, mouse bite, open circuit, spur, and spurious copper as illustrated in Fig. 3.

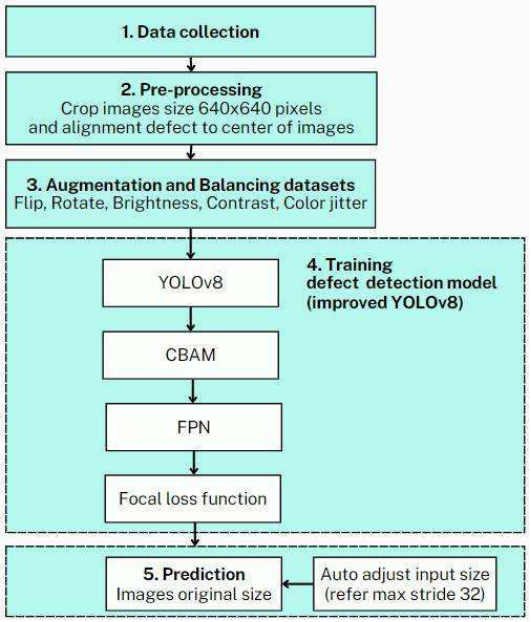


Fig. 1. Our proposed methodology based on improved YOLOv8

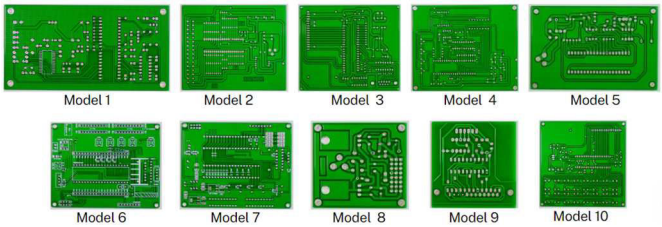


Fig. 2. An example of the ten PCB models from the PKU-Market-PCB dataset.

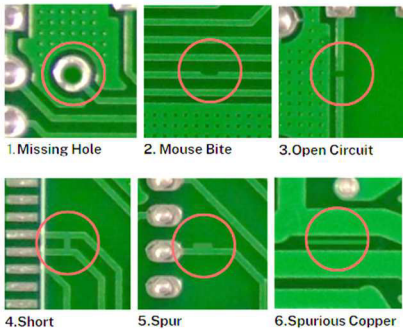


Fig. 3: An example of the six types of defects: missing hole, mouse bite, open circuit, spur, and spurious copper.

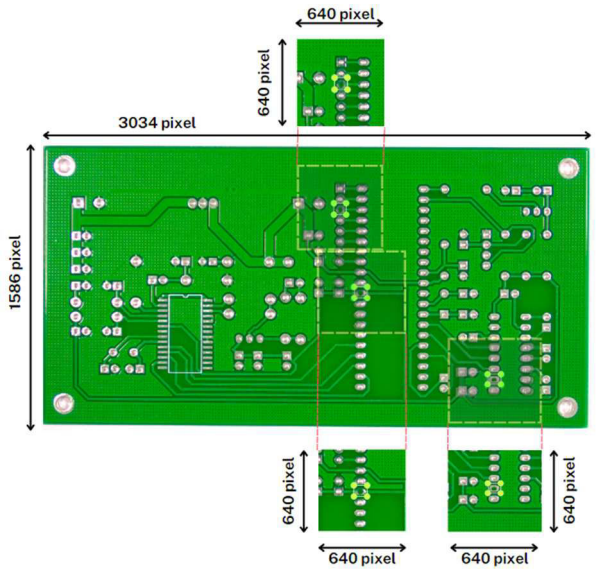


Fig. 4: An example of defect-centered images, each cropped to 640x640 pixels, derived from the original size image.

TABLE I: BALANCING ORIGINAL NUMBER OF DEFECT DATASETS

Class Name	Number of PCB Original size (Images)				Number of defect-centered images (images)			
	train	*valid*	*test*	*Total*	*train*	*valid*	*test*	*Total*
Missing hole	81	12	22	115	351	52	94	497
Mouse bite	79	13	23	115	342	54	96	492
Open circuit	81	12	23	116	339	49	94	482
Short	82	12	22	116	345	51	95	491
Spurious copper	81	12	23	116	351	52	100	503
spur	80	12	23	115	340	51	97	488
Total	484	73	136	693	2068	309	576	2953
% Data sharing					70%	10%	20%	100%

TABLE II: NUMBER OF DEFECT-CENETERED IMAGES FOR TRAINING, VALIDATION, AND TESTING

Datasets	Train	Validation	Test	Total
Original Defect-centered images	2068	309	576	2953
Defect-centered images with Augmentation	1538	220	438	2196
Total	3606	529	1014	5149

B. Defect-Center Image Generation

The pre-processing steps in this research are crucial for preparing the PCB images for training the enhanced YOLOv8 model. These steps include:

a) Annotation Adjustment: To ensure compatibility with YOLO, XML files are converted into YOLO format. The LabelImg tool is used to inspect and correct images with incomplete bounding boxes by expanding them to fully encompass the defect.

b) Defect-centered Image Generation: This research emphasizes preserving the essential features of defects that might be loss during resizing. Instead, a single defect per image is generated with the same input size for YOLOv8 (640x640 pixels). By centering the defect, the entire image is then cropped to 640x640 pixels, as illustrated in Fig. 4.

c) Dataset splitting: The 640x640 pixel images are randomly split into training, validation, and test datasets, ensuring a balanced distribution of defect types. The approximate split ratio is 70% for training, 10% for validation, and 20% for testing in Table I.

C. Data Augmentation and balance dataset

Even though the defect dataset contains recurring patterns, orientations, and characteristics, some images exhibit distinct variations. Additionally, other features such as defect severity, sharpness, and brightness levels vary. Therefore, this research considers augmenting the dataset to increase its diversity while maintaining the balance of each defect type in Table II. The augmentation methods applied include:

a) Flipping: Both vertical and horizontal flips

b) Rotation: Random rotation angles.

c) Brightness and Contrast: Random adjustments to brightness and contrast levels.

d) Color Jitter: Adjustments to hue and saturation using color jitter techniques.

D. Training the Defect Detection Model

a) YOLOv8

YOLOv8 [6] is one of the popular models used for object detection, known for its continuous development and high popularity due to its rapid processing capabilities and high accuracy. It is well-suited for real-time applications and can efficiently detect multiple types of objects within a single image. YOLO (You Only Look Once) was first introduced in 2015 by Joseph Redmon et al., utilizing a One Stage Detector approach for swift processing. The model has undergone continuous enhancements, leading to the development of YOLOv8 in 2023. The key components of YOLOv8 are as follows:

Backbone: YOLOv8 utilizes Cross Stage Partial Darknet (CSPDarknet) as its backbone, which consists of layers designed to extract features from images. The backbone is divided into multiple stages, with each stage performing downsampling to enhance feature levels. It begins with a Stem layer, a ConvModule that reduces the image size by half (Stride=2). This is followed by Stage layers 1 to 4, where each stage includes ConvModules and CSP 2Conv layers.

Neck: The Neck component employs a Path Aggregation Network (PANet), which enhances the efficiency of feature fusion from multiple layers of the Backbone. PANet utilizes both top-down and bottom-up pathways to aggregate features, incorporating the Feature Pyramid Network (FPN) to merge features from different scales.

Head: The Head is the core component for object detection, consisting of multiple layers used for calculating bounding boxes and classification. It features a Decoupled Head that separates the processing of bounding box calculations from object classification.

Loss Function: The loss function comprises Bbox loss for the accuracy of bounding box positions (using GIoU and DFL) and Cls loss for calculating the accuracy of object classification (using BCE loss). The combination of both Bbox loss and Cls loss enables the YOLOv8 model to

b) Convolutional Block Attention Module (CBAM)

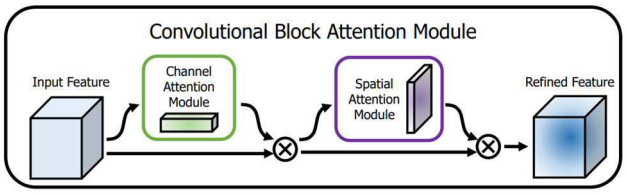


Fig. 5: Convolutional Block Attention Module (CBAM) Architecture [10]

efficiently and accurately detect and classify objects within an image.

This module is designed to enhance the performance of deep neural networks by utilizing the attention mechanism. It is incorporated into the convolutional layers to allow the model to focus more on the important parts of the image. The module comprises two main components: the Channel Attention Module, which helps the model focus on important feature maps within each channel, and the Spatial Attention Module, which enables the model to focus on important positions within the image in Fig. 5.

Incorporating the Convolutional Block Attention Module (CBAM) into the YOLOv8 model structure at each stage enhances defect detection performance. It significantly improves precision, recall, and mAP50 scores.

c) Feature Pyramid Network

Feature Pyramid Networks (FPN), introduced in 2017, are designed to address the challenge of detecting objects of varying sizes within an image. FPN comprises the following pathways in Fig. 6:

Bottom-Up Pathway, this involves computing features through successive convolutional layers of the network (ConvNet), creating feature maps that are scaled down by a factor of 2 at each step.

Top-Down Pathway, this pathway generates higher resolution feature maps by upsampling the coarser feature maps by a factor of 2. These upsampled maps are then merged

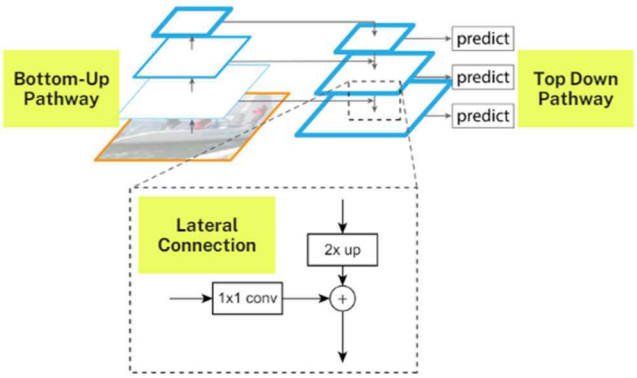


Fig. 6: The structure of Bottom-Up, Top Down and Lateral Connection [7]

with the corresponding feature maps from the Bottom-Up Pathway through lateral connections.

Lateral connections use 1x1 convolutions to reduce the channel dimensions to match those of the higher resolution feature maps. An element-wise addition is performed to combine features from both pathways. This process is repeated iteratively, resulting in feature maps that are both high-resolution and semantically meaningful at all levels of the pyramid. This multi-level feature representation enhances the model's performance in object detection

a) Focal loss function

Focal loss [8] is a loss function designed to improve the learning process of object detection models, particularly in scenarios with class imbalance. In the context of object detection, it is common to encounter situations where objects are either sparse or significantly varied in number across different classes. Focal loss helps reduce false alarms (false positives, FP) in some classes by decreasing the weight of easy examples and increasing the weight of hard examples.

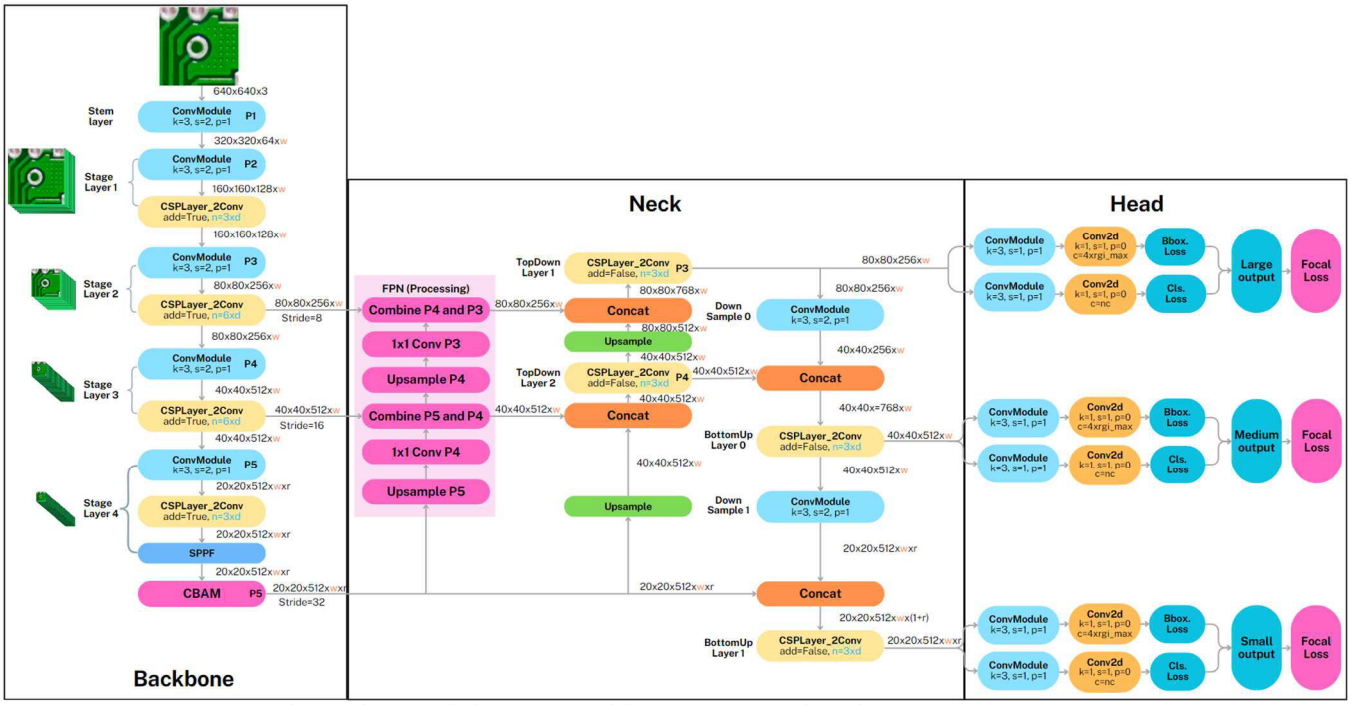


Fig. 7: YOLOv8 Architecture with Convolutional Block Attention Module (CBAM), FPN and Focal Loss

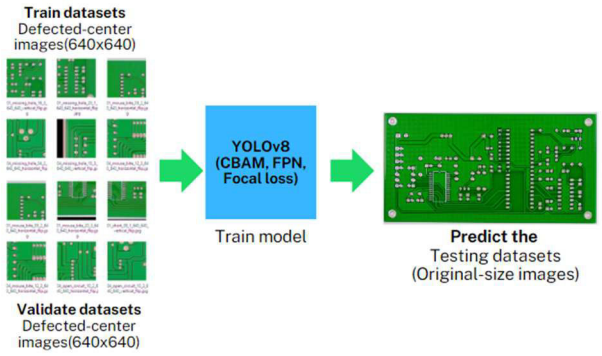


Fig. 8. Image sizes used for training, validation, and prediction

This approach in Fig. 7 enhances the model's performance in detecting defects of various sizes more accurately and comprehensively, especially in cases with high false positives.

E. Prediction

In contrast to the training and validation datasets, which utilize generated centered-defect images, the prediction phase employs the original image size, as illustrated in Fig. 8. During this phase, the program automatically adjusts the width and height slightly, adhering to a specified stride condition that does not exceed 32. The validation dataset comprises 73 large PCB images with a total of 309 defect positions, which are used to evaluate the model's performance.

F. Performance Evaluation

To evaluate the performance of our proposed defect detection model, we measure precision, recall, and average precision (AP) using the intersection over union (IoU) metric to assess the accuracy of bounding box predictions. The AP is divided into two main categories: AP at IoU threshold of 0.5 (AP50), which considers a prediction correct if the IoU is 0.5 or higher.

IV. Experimental Results

A. Parameter Settings

Table III outlines the parameters utilized in conducting the experimental results.

TABLE III: Parameter Settings

Parameter	Setting Value
Baseline Model	YOLOv8n
Optimizer	Adam
Learning Rate	0.001
Batch Size	16
Epochs	100
Input Size	640 pixels
Confidence Threshold	0.5

B. Overall result

Table IV presents the experimental results on the performance evaluation of PCB defect detection using three relevant metrics: precision, recall, and mAP50. Our proposed method achieved highly satisfactory results, with recall and mAP50 scores of 0.996 and 0.99, respectively. Furthermore, our method outperforms others in terms of recall and mAP50 while maintaining comparable precision. This improvement may be attributed to the incorporation of the Convolutional Block Attention Module (CBAM) to emphasize important features, the Feature Pyramid Network (FPN) to enhance multi-scale object detection, Focal Loss to address data imbalance and improve small object detection, and the use of defect-centered training images.

When evaluating performance across defect classes using the mAP50 metric, our method consistently delivers the best results for all six categories of defect types. Specifically, our model achieves a perfect mAP50 score of 1 for nearly all defect types, with the exception of the open circuit defect, which still achieves a highly satisfactory mAP50 score of 0.989, as shown in Table V.

A comprehensive analysis of the per-class performance of our model is presented in Table VI. The results indicate that our model achieves high precision, nearing a score of 1, and demonstrates nearly perfect recall. Specifically, the model exhibits only one misclassification (a false negative) in identifying open circuit defects. However, the model generates some false positives (FP) where background elements were incorrectly classified as defects. Particularly, in the categories of open circuit and spurious copper defects, our model produces 12 and 6 false positive instances, respectively. This misclassification may be attributed to the ambiguity between defects and the background. Further investigation will be conducted to mitigate the false positive classification and enhance the accuracy of our model.

Although, our model is trained on single defect-centered images with variety of data augmentation. Our model exhibits the strong performance in detecting various types and locations of the defects on a single PCB, as demonstrated in Fig. 9. The first and third images illustrate the model's capability to detect multiple classes of defects in different locations on the PCB, extending beyond the single defect-centered image. The second image showcases the detection of a single defect type across various locations on the PCB.

TABLE IV: Performance comparison on PKU-Market-PCB dataset

Method	Precision	Recall	mAP50
YOLO-MBBi (YOLOv5)	**0.950**	0.946	0.953
YOLO-HMC (YOLOv5)	-	0.954	0.985
TDD-YOLO (TVTO&C_0.8)(YOLOv7)	-	0.950	0.978
Proposed: YOLOv8n + CBAM+FPN + Focal Loss (defect-centered image + augmentation)	0.933	**0.996**	**0.998**

The values in bold indicate the best performance for each metric.

TABLE V: COMPARISON OF MAP50 ACROSS DEFECT CLASSES

Model	Missing hole	Mouse bite	Open circuit	Short	Spur	Spurious copper
YOLO-MBBi (YOLOv5)	0.976	0.945	0.973	0.921	0.945	0.957
YOLO-HMC (YOLOv5)	0.995	0.976	0.981	0.979	0.99	0.989
TDD-YOLO (TVTO&C_0.8) (YOLOv7)	0.996	0.968	0.951	0.994	0.975	0.987
Proposed: YOLOv8n + CBAM+FPN + Focal Loss (defect-centered image + augmentation)	**1.00**	**1.00**	**0.989**	**1.00**	**1.00**	**1.00**

The values in bold indicate the best performance for each metric.

TABLE VI: PER-CLASS EVALUATION OF THE PROPOSED MODEL

Defect Type	TP	TN	FP	FN	Precision	Recall	mAP50
Missing hole	52	0	2	0	0.963	1.00	1.00
Mouse bite	54	0	0	0	1.00	1.00	1.00
Open circuit	48	0	12	1	0.80	0.980	0.989
short	51	0	2	0	0.962	1.00	1.00
spur	51	0	0	0	1.00	1.00	1.00
Spurious copper	52	0	6	0	0.897	1.00	1.00
Total	308	0	22	1	0.933	0.996	0.998

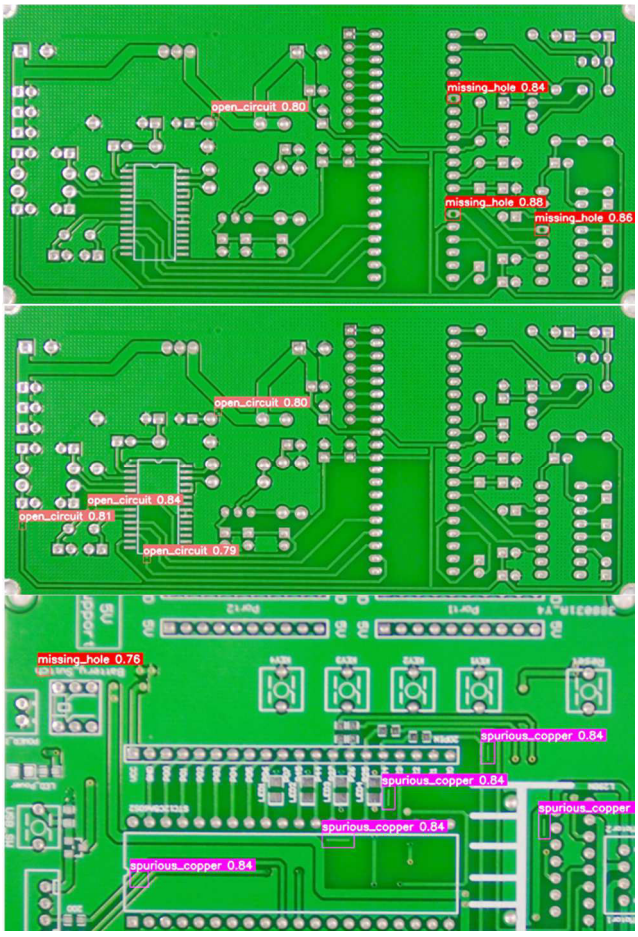


Fig. 9: Examples of prediction result from our model on original size of PCB images

V. CONSLUSION

This paper proposes an effective method for detecting various defect types, which are often small objects, on large PCBs by incorporating single defect-centered image generation with an improved YOLOv8 model. To diversify the defect characteristics on the generated defect-centered images, various data augmentation techniques are utilized. To enhance YOLOv8's performance in detecting small defects, several modifications are made: a Convolutional Block Attention Module (CBAM) layer is added to focus on important extracted features, a Feature Pyramid Network (FPN) is employed to improve multi-scale defect detection, and a Focal Loss function is applied to address defect imbalance in the generated data. During the classification step, original-size defect images are used, or they are slightly adjusted to fit the stride-32 condition. Experimental results on the benchmark dataset demonstrate that our proposed method is capable of detecting nearly all defects while producing only a few false positives in some defect types. Additionally, our method outperforms others in terms of recall and mAP50 while maintaining comparable precision. To further mitigate the effect of false positive classifications, additional investigation will be conducted to enhance the model's accuracy.

REFERENCES

[1] J. Redmon and A. Farhadi, "YOLO-MBBi: An Enhanced YOLOv5 Model for PCB Surface Defect Detection," in *Proceedings of the 2023 International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 123-130.
[2] H. Wang, Y. Li, and Z. Zhang, "RAR-SSD: Improved Fault Recognition Model for PCB Manufacturing," IEEE Trans. Ind. Electron., vol. 70, no. 6, pp. 1123-1135, Jun. 2023.
[3] S. Lee, D. Kim, and J. Park, "Lightweight Neural Network-based Real-time PCB Defect Detection System," in *Journal of Manufacturing Systems*, vol. 61, pp. 45-55, Dec. 2023.
[4] M. Chen, K. Liu, and J. Wu, "YOLO-HMC: An Improved Method for PCB Surface Defect Detection," IEEE Trans. Autom. Sci. Eng., vol. 21, no. 1, pp. 98-108, Jan. 2024.
[5] T. Zhang, L. Wang, and Q. Liu, "An Efficient Tiny Defect Detection Method for PCB With Improved YOLO Through a Compression Training Strategy," IEEE Access, vol. 12, pp. 2024-2036, Apr. 2024.
[6] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788). IEEE. https://doi.org/10.1109/CVPR.2016.91
[7] T.-Y. Lin et al., "Feature Pyramid Networks for Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 936-944. doi: 10.1109/CVPR.2017.106.
[8] T.-Y. Lin et al., "Focal Loss for Dense Object Detection," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2017, pp. 2980-2988.
[9] Y. Liu, Z. Zhang, H. Wang, and J. Li, "PCB Defect Detection via Local Detail and Global Dependency Information," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 4390-4401, Sep. 2023. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10538067
[10] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.

turnitin Page 168 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Detection of Infective Juvenile Stage of Entomopathogenic Nematodes Using Deep Learning

Uthai Phuyued
College of Innovative Technology and Engineering,
Dhurakij Pundit University,
Bangkok, Thailand
Email: 65130132@dpu.ac.th

Thanapat Kangkachit
College of Innovative Technology and Engineering,
Dhurakij Pundit University,
Bangkok, Thailand
Email: thanapat.kan@dpu.ac.th

Duangjai Jitkongchuen
Manpower Development Division
Big Data Institute
(Public Organization)
Bangkok, Thailand
Email: duangjai.ji@bdi.or.th

***Abstract*— Entomopathogenic nematodes, particularly *Steinernema glaseri*, are essential for biological pest control, with their infective juvenile (IJ) stage being critical for insect infection. Accurate identification of the IJ stage, typically performed using a stereomicroscope, is time-consuming and prone to human error. This research proposes a deep learning-based method for identifying the IJ stage nematodes from stereomicroscope images. The YOLOv5s architecture is utilized and enhanced by image pre-processing with Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve image quality. Experimental results confirm the effectiveness of our approach, achieving a precision of 0.781, recall of 0.783, and a mean Average Precision (mAP50) of 0.817. These results demonstrate the capability of the deep learning model combined with CLAHE to enhance detection accuracy, reduce processing time, and minimize errors.**

***Keywords*— Deep Learning, YOLOv5, CLAHE, Steinernema glaseri, Infect Juvenile Stage**

I. Introduction

Entomopathogenic nematodes (EPNs) in the families Steinernematidae and Heterorhabditidae, belonging to the order Rhabditida, are small roundworms that cause diseases in insects.[1][2] Several species are beneficial in agriculture for controlling insect pests, such as Steinernema carpocapsae, S. siamkayai, S. riobrave, and S. glaseri, among others. The third larval stage of nematodes in both families, known as the infective juvenile (IJ) stage, is the only stage capable of infecting host insects. IJ nematodes enter the body cavity of insects by penetrating the cuticle directly or through natural openings such as the mouth, spiracles, and anus.[2][3] Once inside, the nematodes release symbiotic bacteria into the host's body cavity, causing septicemia and killing the host within 24-48 hours. Steinernema nematodes work in conjunction with bacteria from the genus Xenorhabdus.[4]

S. *glaseri* nematodes have shown potential in controlling beetle larvae (white grubs) in the order Coleoptera, which cause significant agricultural damage. Consequently, entomopathogenic nematodes are being developed as biocontrol agents in many countries, including Thailand. Commercial rearing of EPNs is currently underway, but selecting the infective juvenile (IJ) stage under a stereomicroscope is challenging and requires skilled researchers, leading to time and resource inefficiencies.

To address these challenges, computer vision and AI are being utilized to reduce processing times across various tasks. Recent research has leveraged AI techniques to enhance object detection capabilities. For instance, the YOLO (You Only Look Once) [9] architecture has been employed to detect and classify small and complex objects in various domains, demonstrating notable improvements in precision and speed. Studies have also highlighted the effectiveness of image pre-processing techniques, such as Contrast-Limited Adaptive Histogram Equalization (CLAHE) [5], in enhancing the visibility of details in images. CLAHE improves contrast in homogeneous areas of an image without adding excessive noise, which is crucial for detecting subtle features in nematode images.

This study aims to develop a robust system for identifying the IJ stage of *S. glaseri* using stereomicroscope images, incorporating CLAHE for image enhancement and YOLO for object detection to optimize accuracy and efficiency[10-14]. The remainder of this paper is organized as follows: Section 2 reviews related works on nematode detection and image processing techniques. Section 3 details the methodology. Section 4 presents the experimental setup and results, while Section 5 discusses the findings and concludes the study.

II. Related Works

The development of computer vision and artificial intelligence (AI) through deep learning technology has become crucial in distinguishing various types of photographic images. Microscope images, in particular, are among those that greatly benefit from the application of these technologies, allowing us to gain a deeper understanding and analysis of our world. In this chapter, we will explore the fundamental concepts and intriguing applications of these technologies in the context of classifying microscope images, focusing on enhancing image analysis efficiency and effectiveness across different fields. In this experiment, we have also incorporated the technique of Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the performance of classifying microscopic images of small organisms.

1) Contrast Limited Adaptive Histogram Equalization (CLAHE) [5]

CLAHE is a noise-reduction method derived from Adaptive Histogram Equalization (AHE). CLAHE limits the contrast enhancement in homogeneous areas by clipping the original histogram and redistributing the clipped pixels equally across the histogram's grayscale levels. Unlike a standard histogram, the CLAHE method allows users to set a maximum pixel intensity threshold, providing more control over the density of pixel values.

Principles of CLAHE Operation:

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

1. Divide the Image into Sub-grids (Tiles): The image is divided into smaller grids, such as 8x8 or 16x16, where each grid is processed separately.

2. Calculate Histogram for Each Sub-grid: A histogram is created for each grid, representing the distribution of pixel intensity values within the grid.

3. Limit Contrast (Clip Limit): A maximum frequency limit is set for the histogram of each grid. If the frequency of any intensity value exceeds this limit, the excess frequency is redistributed to other intensity values in the histogram. The clip limit helps prevent excessive contrast enhancement that could introduce noise into the image.

4. Create Cumulative Distribution Function (CDF): After limiting the contrast, a new CDF is generated for the histogram to adjust the pixel intensity values within the grid.

5. Perform Interpolation: Once the pixel intensity values in each grid are adjusted, interpolation is performed between the grids to ensure smooth and natural contrast transitions across the image.

2) Nematode Identification using Artificial Neural Networks [6].

This research investigates the classification of nematode species using sketch images for taxonomic data entry, comparing them with images of nematodes under a microscope. In this study, the Xception model[16] was chosen from a comparison of 13 models for nematode species classification. The dataset consists of nematode images captured using a light microscope at 100x magnification. The experimental results showed an accuracy of 88.28% for the IJ stage and 69.45% for the adult stage.

3) Seagrass Blurred Image Enhancement and Detection using YOLO and CLAHE Algorithms Performance Comparison[7]

This research investigated the effectiveness of Contrast Limited Adaptive Histogram Equalization (CLAHE) and the You Only Look Once (YOLO) algorithm in enhancing object recognition within blurred fluorescent images of underwater ecosystems, with a specific focus on seagrass. The study highlighted that applying CLAHE to the images significantly improved YOLO-V5's accuracy, leading to an enhancement of 8% to 10%. The research utilized two versions of the YOLO model, YOLO-V3 and YOLO-V5, to compare their performance. In terms of effectiveness, YOLO-V3 was found to be the most suitable algorithm for this research. However, YOLO-V5 demonstrated the ability to detect almost all blurred images.

III. Methodology

In the task of distinguishing the stages of nematodes under a microscope, laboratory personnel must possess significant expertise to differentiate between the IJ stage and non-IJ stage nematodes. To facilitate this task, reduce the time required, and increase the accuracy of IJ stage classification, deep learning techniques have been applied. Additionally, image enhancement techniques have been employed to highlight the distinctive features of the objects in the images. Specifically, the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique has been incorporated into the preprocessing of images obtained from the microscope.[10]

A. Data collection

The images were captured using a Digital Single Lens Reflex (DSLR) camera through a stereo microscope. These images contain nematodes at both the IJ stage and non-IJ stage mixed within the same image, as shown in Fig 1. The prominent difference between the two classes is that, at the IJ stage, the nematodes are encased in a transparent sheath, as depicted in Fig 2, whereas non-IJ stage nematodes lack this sheath, as illustrated in Fig 3. To collect the data, equipment was set up to capture images of nematodes and differentiate the IJ stage. A DSLR camera was used to capture images through the microscope lens, with a USB to mini-USB cable connecting the DSLR camera to the computer, as shown in Fig 4. The details of the collected sample images are presented in Table I.

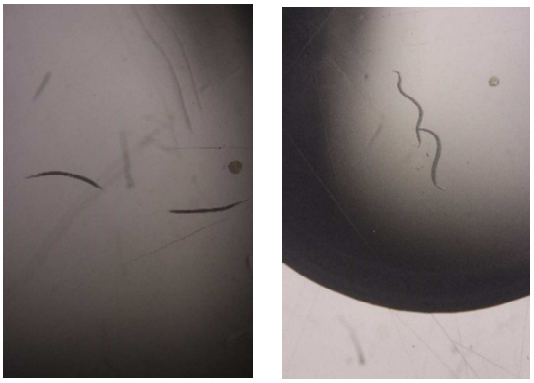


Fig. 1. Nematode image captured with a microscope at 10x magnification.



Fig. 2. Images of Nematodes at the IJ Stage.

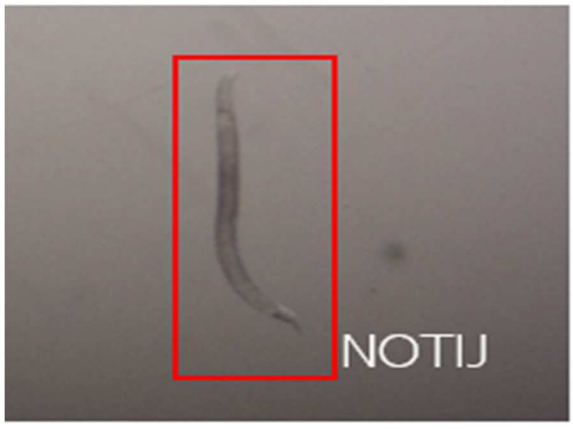


Fig. 3. Images of Nematodes at the Non-IJ Stage



Fig. 4. Installation of Equipment for Capturing Images of Nematodes

TABLE I: EQUIPMENT SETTING

List	VALUE
Microscope Magnification	10x
Image Size	3456 × 5184 pixels
Image Type	Color
Number of Images	788

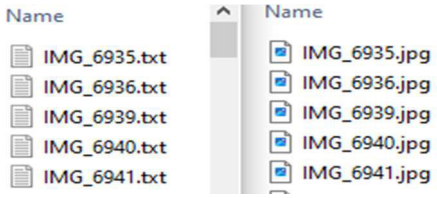


Fig. 5. The label files generated from the labeling process using the labelImg Program

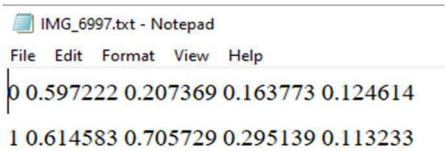


Fig. 6. Details contained within the label file.

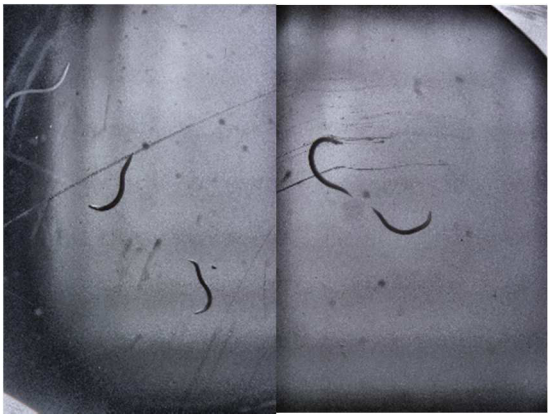


Fig. 7. Image after applying CLAHE.

B. *Labeling of images captured by the microscope*

In image detection tasks, labeling involves specifying the location of objects. The tool used for this purpose is "labelImg", which provides outputs compatible with YOLOv5. The labeling process generates a text file with labels, where the file name must match the image file name, as shown in Fig 5. The contents of this text file, demonstrated in Fig. 6., are as follows:

- The first value (1,0) denotes the type of object, 0 denoted non-IJ stage and 1 denoted IJ stage.
- The second and third values (e.g. 0.232639 0.707562 in the first row, 0.785735 0.376447 in the second row) represent the coordinates of the center of the object within the image, in X and Y format.
- The fourth and fifth values (e.g. 0.300926 0.111883 in the first row, 0.217882 0.051505 in the second row) indicate the width and height of the object within the image, respectively, in Width and Height format.

C. *Image Pre-processing*

Image pre-processing is a critical step in object detection tasks, as it enhances image quality and improves the performance of deep learning models. Contrast Limited Adaptive Histogram Equalization (CLAHE) is an effective technique in this regard, as it enhances image contrast and makes key features more distinguishable [11][15]. In the image pre-processing stage utilizing CLAHE, the images are adjusted using parameters of clipLimit=5.0 and tileGridSize=(8, 8), as illustrated in Fig. 7.

D. *Training Model*

After completing the image pre-processing step using CLAHE, the images are divided into training and validation datasets, as indicated in Table II. The model is then trained using the YOLOv5 architecture to capture the characteristics of IJ and NonIJ stages from the training dataset. Detailed information on YOLOv5 is provided in the following sections.

YOLOv5[8] has been optimized for faster and more accurate object detection, making it highly efficient for real-time applications. It includes various pre-trained models that can be fine-tuned for specific tasks, reducing the time and computational resources needed for training from scratch. Advanced augmentation techniques and hyperparameter optimization further enhance the model's robustness and accuracy. YOLOv5's combination of speed, accuracy, and flexibility underscores its position as a leading tool for object detection.

The four main components of YOLOv5, depicted in Fig. 8, are as follows:

1. Backbone: Responsible for extracting features from the original image, using CSPDarknet53 to maximize feature extraction efficiency.
2. Neck: Powered by PANet, it aggregates features for multi-scale object detection and complex scenes, ensuring the Head receives informative and rich features.
3. Head: The final stage of the detection pipeline, predicting bounding boxes and object classes, translating features into actionable outputs.

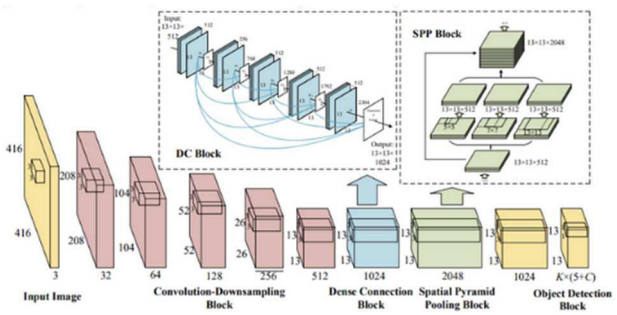


Fig. 8. YOLO Architecture

4. Loss function: Integrates Localization Loss, Confidence Loss, and Classification Loss to evaluate and optimize the model's performance, enhancing accuracy and reliability.

E. Performance Evaluation

Our model is evaluated using three key metrics: precision, recall and mean average precision (mAP).

Precision reflects the accuracy of the positive predictions made by the model, indicating how many of the predicted positive instances are actually correct. The formula for precision is:

$$P = \frac{TP}{TP+FP} \quad (1)$$

where TP and FP are the number of true positives (TP) and false positives (FP) respectively.

Recall evaluates the model's ability to identify all relevant instances, reflecting how many of the actual positive instances are correctly detected. The formula for precision is:

$$R = \frac{TP}{TP+FN} \quad (2)$$

where FN denotes the number of false negatives (FP).

Mean Average Precision (mAP) provides a comprehensive measure of the model's overall performance across multiple classes and various levels of prediction confidence. mAP is the average of the Average Precision (AP) values calculated from the precision-recall curve at different thresholds for each class. It is given by:

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad (3)$$

where n is the total number of classes and AP_i is the Average Precision for class i.

IV. Experimental Results

A. Experimental Setup

After completing the image pre-processing steps, the processed images are divided into training and validation datasets. The training dataset is balanced in terms of the number of IJ and NonIJ classes. However, the validation dataset contains a higher number of IJ labels compared to NonIJ labels. Details of the entire dataset are provided in Table II.

Table III summarizes the parameters used for conducting the experiments. We selected YOLOv5s, which is designed to offer a balance of speed and performance while maintaining high accuracy in object detection. This variant is notably smaller in size compared to other versions, making it more efficient for our purposes.

B. Overall result

For the experiments where original images were used for both training and testing, the model was more effective at distinguishing nematodes in the IJ stage compared to those in the non-IJ stage. Nonetheless, it still managed to classify non-IJ stages reasonably well. Furthermore, the model demonstrated the capability to work with images processed using Contrast Limited Adaptive Histogram Equalization (CLAHE), as evidenced by Table IV, where precision, recall, and mAP values remained consistent across the board.

When the model was trained and tested with images processed using Contrast Limited Adaptive Histogram Equalization (CLAHE), it showed enhanced capability in identifying IJ-stage nematodes compared to standard images. The CLAHE-processed images improved precision and mAP values, indicating a better detection of subtle features. However, the model exhibited a decrease in accuracy for non-IJ stages, with a higher rate of misclassification where non-IJ stages were incorrectly identified as IJ stages. Despite this, Table IV highlights that the precision and mAP values were notably improved with CLAHE-processed images, although recall was less robust compared to standard images.

In contrast, when using images adjusted with automatic contrast and brightness settings, the model failed to effectively detect nematodes. This failure suggests that the automatic adjustment did not achieve the necessary image quality for successful training and testing, leading to suboptimal performance. This emphasizes the importance of appropriate image pre-processing techniques in optimizing model accuracy.

TABLE II: NUMBER OF IMAGES AND LABELS

	Training Dataset	Validation Dataset
Number of Images	788	106
Total Number of Labels	1135	177
Number of Labels for Class "IJ"	568	147
Number of Labels for Class "NonIJ"	567	30

TABLE III: PARAMETER SETTINGS FOR CLAHE AND YOLOV5

CLAHE	
clipLimit	5.0
tileGridSize	8,8
YOLOv5s	
epochs	20
Image size	640*640
batch-size	16
Learning rate	0.01
optimizer	SGD
Validate	10%

TABLE IV: TEST RESULTS COMPARED TO ALTERNATIVE METHODS

Method	Precision	Recall	mAP50	mAP50:95
Train and Test with original images	0.737	**0.843**	0.805	0.528
Train with original images, Test with CLAHE-processed images	0.737	**0.843**	0.805	0.528
Train with CLAHE-processed Images, Test with original Images	0.621	0.713	0.702	0.44
Train with images Processed using automatic contrast and brightness adjustment, Test with original images	0.631	0.409	0.555	0.331
Train and Test with images processed using automatic contrast and brightness adjustment	0	0	0	0
Train and Test with CLAHE-processed Images	**0.781**	0.783	**0.817**	**0.531**

The values in bold indicate the best performance for each metric.

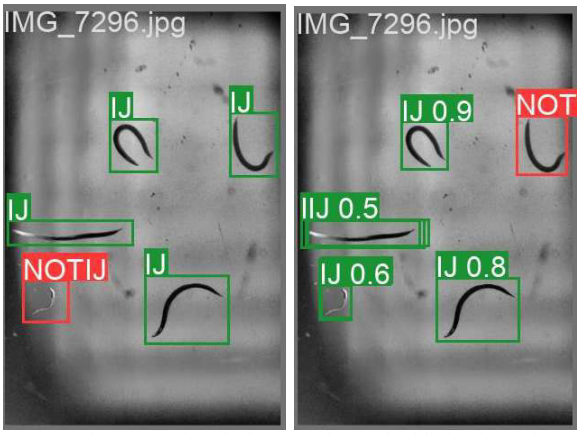


Fig. 9. Examples of misclassified nematodes are highlighted in red.

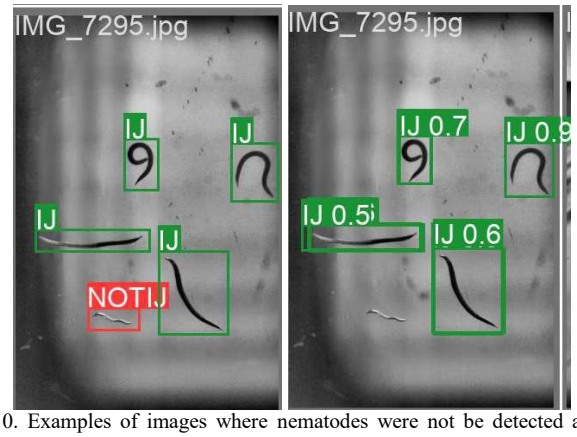


Fig. 10. Examples of images where nematodes were not be detected are highlighted in red.

In addition to evaluating performance metrics, we also examined the test images and identified cases where the predicted class did not align with the correct class, as illustrated in Fig 9. In certain instances, the model failed to detect the desired objects altogether. These issues were particularly evident in images that were similar in content but differed in the positional location of the objects, as depicted in Fig 10.

The discrepancies observed in Fig 9 highlight instances of incorrect class prediction, suggesting potential shortcomings in the model's classification accuracy. Meanwhile, Fig 9 demonstrates cases where the model could not detect objects despite their presence, which could be attributed to factors such as occlusion or suboptimal feature extraction.

V. CONCLUSION

This research presents a deep learning-based approach for the precise identification of the infective juvenile (IJ) stage of *Steinernema glaseri* nematodes using stereomicroscope images. Traditional methods of IJ stage identification rely heavily on manual stereomicroscopy, which is both labor-intensive and susceptible to human error. Our approach leverages the YOLOv5s architecture and enhances it with Contrast Limited Adaptive Histogram Equalization (CLAHE) for improved image quality. The integration of CLAHE significantly boosts the model's performance, achieving a precision of 0.781, recall of 0.783, and a mean Average Precision (mAP50) of 0.817. These results highlight the effectiveness of our method in accurately detecting the IJ stage, reducing processing time, and minimizing errors compared to conventional techniques. Overall, this research demonstrates that deep learning, when combined with advanced image pre-processing, offers a robust solution for enhancing the efficiency and accuracy of nematode identification in biological pest control. Future studies could benefit from experimenting with other deep learning architectures and preprocessing techniques. This would provide a broader understanding of how different models and image enhancement methods impact the detection and classification performance in various biological imaging contexts.

REFERENCES

[1] Kaya, H. K., and R. Gaugler. 1993. Entomopathogenic nematodes. Ann. Rev. Entomol. 38:181–206.

[2] Poinar Jr., G. O. 1990. Taxonomy and biology of Steinernematidae and Heterorhabditidae. pp.23–61. In: R. Gaugler, and H. K. Kaya. Nematodes in Biological Control. CRC Press, Boca Raton, FL.

[3] Grewal, P. S., R. U. Ehlers, and D. I. Shapiro-Ilan. 2005. Nematodes as Biocontrol Agents. CABI Publishing, Wallingford, UK.

[4] Ehlers, R. U., S. Linau, O. Krasomil, and K. H. Osterfield. 1998. Liquid culture of entomopathogenic nematode bacterium complex (Heterorhabditis megidis Poinar, Jackson & Klein)/(Photorhabdus luminescence) Bio. Control. 43(1): 77-88.

[5] Karel Zuiderveld "Contrast Limited Adaptive Histogram Equalization,"[online].https://www.cse.unr.edu/~bebis/CS474/Student PaperPresentations/1994%20-%20CLAHE.pdf. (Accessed: May 15, 2024,).

[6] Jason Uhlemann, Oisin Cawley and Thomais Kakouli-Duarte. 2019. Nematode Identification Using Artificial Neural Networks. Environ 2019: Engagement for Climate Action, 29th Colloquium of Irish Environmental Research.

[7] Elmo Ranolo, Archival Sebial, Anthony Ilano and Angie Ceniza Canillo. 2023. Seagrass Blurred Image Enhancement and Detection using YOLO and CLAHE Algorithms Performance Comparison. 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT).

[8] Xingkui Zhu, Shuchang Lyu, Xu Wang and Qi Zhao. 2021. You Only Look Once: Unified, Real-Time Object Detection. Computer Science, Computer Vision and Pattern Recognition. Volume 26 Aug 2021.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779-788). IEEE. https://doi.org/10.1109/CVPR.2016.91.

[10] B. Li, K. Serrano, M. Mazzaro, M. Wu, W. Wang and M. Zhu, "Identification of Cyanobacteria for Harmful Algal Blooms Research Using the YOLO Framework," *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2023, pp. 0407-0415, doi: 10.1109/UEMCON59035.2023.10316078.

[11] W. A. K. Adji, A. Amalia, H. Herriyance and E. Elizar, "Abnormal Object Detection In Thoracic X-Ray Using You Only Look Once (YOLO)," *2021 International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE)*, Banda Aceh, Indonesia, 2021, pp. 118-123, doi: 10.1109/COSITE52651.2021.9649500.

[12] L. Guodong, F. Lihui, L. Jihua and Y. Lei, "Underwater image enhancement and detection based on convolutional DCP and YOLOv5," *2022 41st Chinese Control Conference (CCC)*, Hefei, China, 2022, pp. 6765-6772, doi: 10.23919/CCC55666.2022.9902814.

[13] R. -C. Chen, C. Dewi, Y. -C. Zhuang and J. -K. Chen, "Contrast Limited Adaptive Histogram Equalization for Recognizing Road Marking at Night Based on Yolo Models," in *IEEE Access*, vol. 11, pp. 92926-92942, 2023, doi: 10.1109/ACCESS.2023.3309410.

[14] F. Luo, Y. Du, L. Fu, C. Chen and R. Wang, "Object Detection in Harsh Underwater Environment Based on YOLOv5s-CCAA," *2023 2nd International Joint Conference on Information and Communication Engineering (JCICE)*, Chengdu, China, 2023, pp. 26-31, doi: 10.1109/JCICE59059.2023.00016.

[15] I. Lashkov, R. Yuan and G. Zhang, "Edge-Computing-Facilitated Nighttime Vehicle Detection Investigations With CLAHE-Enhanced Images," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13370-13383, Nov. 2023, doi: 10.1109/TITS.2023.3255202.

[16] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

turnitin Page 174 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Effect of Sliding Window Sizes on Sensor-Based Human Activity Recognition Using Smartwatch Sensors and Deep Learning Approaches

Sakorn Mekruksavanich[1], Ponnipa Jantawong[1] and Anuchit Jitpattanakul[2,*]
[1]*Department of Computer Engineering, School of Information and Communication Technology University of Phayao*, Phayao, Thailand
sakorn.me@up.ac.th and ponnipa.jantawong@gmail.com
[2]*Intelligent and Nonlinear Dynamic Innovations Research Center, Department of Mathematics Faculty of Applied Science, King Mongkut's University of Technology North Bangkok*, Bangkok, Thailand
anuchit.j@sci.kmutnb.ac.th

***Abstract*—Smartwatch sensors for human activity recognition (HAR) have gained significant attention due to their applications in healthcare and fitness monitoring. The effectiveness of HAR systems largely depends on the choice of sliding window widths for sensor data segmentation. This study investigates the impact of varying sliding window widths on the accuracy of HAR using wristwatch sensors and deep learning techniques. We conducted experiments using the daily human activity (DHA) dataset, comprising sensor data from 11 distinct activities. Data was preprocessed and segmented using window sizes ranging from 5 to 40 seconds. Four deep learning models (CNN, LSTM, BiLSTM, and CNN-LSTM) were employed and evaluated using accuracy, precision, recall, and F1-score. Window size significantly affected HAR performance. Smaller windows improved short-duration activity recognition but increased computational complexity, while larger windows reduced computational load but decreased accuracy for rapid activity changes. The CNN-LSTM hybrid model consistently outperformed other models, achieving 92.11% accuracy with a 20-second window and overlapping segmentation. This research provides valuable insights into balancing recognition accuracy and computational resources in smartwatch sensor-based HAR, contributing to the development of efficient and accurate systems for real-world applications.**

***Keywords*—human activity recognition, smartwatch sensors, sliding window, deep learning, sensor data segmentation**

I. Introduction

Human activity recognition (HAR) has gained considerable prominence in the past few decades owing to its many uses for medical care [1]–[4], sports monitoring [5]–[8], and individual support [9]–[12]. Due to the progress in wearable devices, smartwatches have gained popularity for gathering sensor data about people's movements. These gadgets are equipped with various sensors, including accelerometers, gyroscopes, and magnetometers, that collect detailed data on the individual's motions and actions [13].

Deep learning algorithms have demonstrated exceptional effectiveness in HAR challenges, surpassing conventional machine learning techniques [14]–[16]. These algorithms, including convolutional neural networks (CNNs) and long-short-term memory (LSTM) networks, can acquire structured characteristics from unprocessed sensor data without requiring any human feature engineering [17]. A vital component of HAR utilizing deep learning involves pre-processing data collected by sensors. Dividing the uninterrupted flow of sensor data into windows of specific proportions is a widely used method to prepare the data for training and inference of models. The sliding window size selection may substantially affect the identification quality and the computing complexity of HAR applications.

Prior research has investigated the impact of different sliding window widths on HAR utilizing diverse sensor paradigms and deep learning algorithms [18], [19]. The study results indicate that employing large sliding windows is optional for achieving outstanding performance measures. This is because doing so would increase the cost of processing and the reaction capabilities of any application developed through the trained model [17]. Nevertheless, a more thorough examination is needed regarding the influence of varying window widths on the utilization of smartwatch detectors and cutting-edge deep learning methods.

This study examines how varying sliding window widths impact the accuracy of sensor-based HAR utilizing wristwatch sensors and deep learning methods. The information we gather is the daily human activity (DHA) dataset. It consists of sensor data collected from individuals engaged in 11 distinct activities, including strolling, jogging, seating, and standing. We operate many algorithms for deep learning, such as CNN, LSTM, bidirectional LSTM (BiLSTM), and CNN-LSTM, to categorize the actions based on the retrieved characteristics. We assess the efficacy of each approach by using criteria including accuracy, precision, recall, and F1-score.

This paper's primary contributions are outlined below:

1) Our study examines how varying sliding window widths impact the accuracy of HAR utilizing wristwatch sensors and advanced deep learning methods.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

2) Our findings provide valuable insights into the practical implications of the trade-off between identification accuracy and computing complexity when determining sliding window widths for sensor-based HAR.

II. Related Works

The process of data segmentation in HAR, particularly the widely used sliding window approach, is a significant area of research 20], [21. This approach, known for its simplicity and reliability, has been the focus of numerous studies [22]. Researchers have explored a variety of window lengths in their investigations, underscoring the importance of this method in HAR.

Movements such as strolling, running, and moving up or down the stairs have been recognized using small window sizes, 0.5 s and 0.8 s [23]. The classification of stationary, walking, running, and biking movement phases has been performed using a decision tree classifier in combination with a window size of 1 second [24]. Furthermore, a time interval of 2 seconds, combined with a neural network [25], has been used to categorize different types of motion, such as strolling, going up and down stairs, running, and seated with different body positions. This approach has yielded a mean accuracy of 93%. Portable smartphones have been utilized for categorizing walking, standing, and ascending stairs with high accuracy scores of 84% utilizing larger window widths, such as 5 seconds, and numerous approaches [26]. The recognition of strolling, immobile, running, and cycling actions is achieved by employing a window size of 7.5 seconds when the smartphone is put in the individual's trouser pocket [27]. The approach, along with the K-nearest neighbor machine learning technique, results in a classification accuracy of 93.9%.

Investigators in deep learning for HAR have examined the effects of segmenting input data into window sizes as a pre-processing technique. Mairittha et al. [28] conducted data annotation for a motion detection technique employing inertial (acceleration and angular velocity) portable sensing in both simple-LSTM and hybrid CNN-LSTM models. The experimenters used a window size of 5.12 seconds (at a frequency of 20 Hz), resulting in about 100 frames with overlapping. Ebner et al. [29] introduced a new method that utilizes analytical conversions and artificially created sensor channels to recognize activities, namely acceleration and rotational velocity. The researchers conducted experiments with window widths of 2, 2.5, and 3 seconds at a sampling rate of 50 Hz. This corresponds to 100, 125, and 150 frames, respectively. The experiments were conducted without any overlapping. The authors' conclusion indicated that the impact of window size on accuracy was minimal, with a modest inclination towards decreased accuracy as window widths increased.

Despite the considerable research on the selection of window widths for HAR, there is a clear need for comprehensive studies that specifically focus on the effect of sliding window sizes when using smartwatch sensors and advanced deep-learning models. This study is designed to meet this need by examining the impact of varying sliding window widths on the accuracy of sensor-based HAR utilizing wristwatch sensors and deep learning methodologies.

III. Methodology

The sensor-based HAR architecture implemented in this study consists of four primary steps: data gathering, data pre-processing, data production, and model training and assessment, as seen in Fig. 1.

A. Daily Human Activity (DHA) Dataset

The DHA dataset [30] from Kookmin University contains smartwatch accelerometer data from two individuals performing 11 distinct activities over four weeks. Data was collected using an Apple Watch Series-2 at 10Hz and transmitted to an iPhone 6. Activities were conducted in office spaces (5), kitchens (3), and outdoors (3), with no concurrent activities assumed. The dataset includes information on the number of samples for each activity. The quantity of samples for each task is condensed in Table I.

TABLE I: A list of activities in DHA dataset

Activity	Abbrv.	Location	No. of data[a]
Office work	Ow	Office	62,711
Reading	Re	Office	36,976
Writing	Wr	Office	27,677
Taking a rest	Tr	Office	31,265
Playing a game	Pg	Office	51,906
Eating	Ea	Kitchen	46,155
Cooking	Co	Kitchen	10,563
Washing dishes	Wd	Kitchen	10,712
Walking	Wa	Outdoors	25,768
Running	Ru	Outdoors	6,452
Taking a transport	Tt	Outdoors	28,483

[a]Number of raw accelerometer data.

Fig. 2 shows 2D graphs of accelerometer data for 11 activities in the DHA dataset, displaying x, y, and z acceleration over time. The patterns vary distinctly across activities:

- Office work, reading, writing, resting, gaming, and eating show modest acceleration amplitudes.
- Walking and running display higher, more regular accelerations.
- Cooking, dish cleaning, and transportation exhibit mild acceleration fluctuations.

B. Data Pre-processing

The raw sensor data was subjected to noise reduction and normalization during the pre-processing step. The pre-processed sensor data was segmented by a fixed-width sliding window technique through completing these procedures.

This research used two segmentation techniques to examine the influence of different sliding window widths. The initial approach, known as the overlapping temporal window (OW), entails the application of a window of a predetermined size to the input data sequence. This window creates training and test samples, employing a particular validation method. Nevertheless, this approach results in substantial bias due

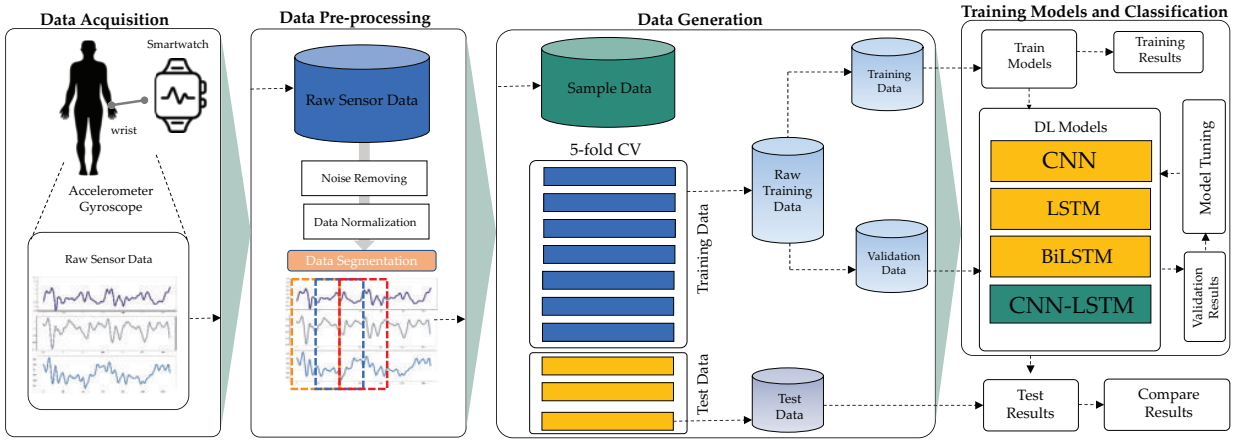


Fig. 1: The HAR framework based on smartwatch sensors used in this work.

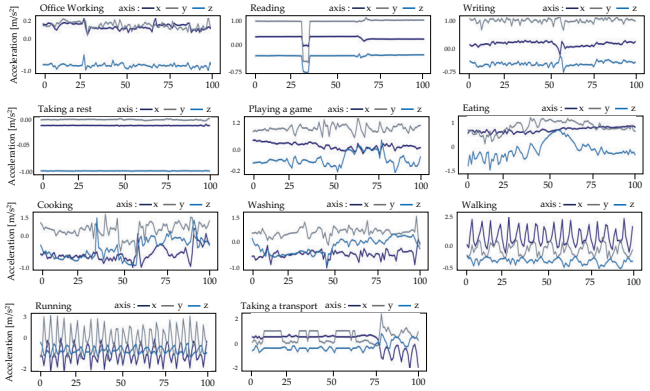


Fig. 2: Some samples of 11 daily human activities from DHA dataset.

to the 50% overlap between subsequent sliding windows. In order to reduce this bias, a different method called the non-overlapping temporal window (NOW) was used. The NOW strategy, as opposed to the OW method, has the drawback of producing fewer samples due to the lack of overlapping temporal frames. Fig. 3 illustrates two sample generation algorithms for segmenting sensor data, where X, Y, and Z correspond to the three portions of a tri-axial IMU sensor.

C. Deep Learning Models

This study examines the impact of different sliding window widths on sensor-based HAR employing smartwatch sensors and Deep Learning methods. This study utilizes four deep learning approaches: CNN, LSTM, BiLSTM, and CNN-LSTM. The structure of each model employed in this study is illustrated in Fig. 4.

IV. Experiments and Results

Our research investigates how varying sliding window sizes affect deep learning models' interpretation in identifying human movements from smartwatch sensor data. We utilized the DHA dataset to conduct our experiments. Four distinct neural network architectures were employed: CNN, LSTM, BiLSTM, and a hybrid CNN-LSTM model.

The experimentation process involved training and evaluating these models with different time windows. We explored durations spanning from 5 to 40 seconds. In our analysis, we applied both non-overlapping and overlapping segmentation approaches to the data. We relied on widely-used performance metrics to gauge the efficacy of each model configuration. These included accuracy, precision, recall, and F1-score. This comprehensive evaluation allowed us to assess the impact of window size on the models' ability to classify human activities based on wearable sensor input accurately.

A. Experimental Findings

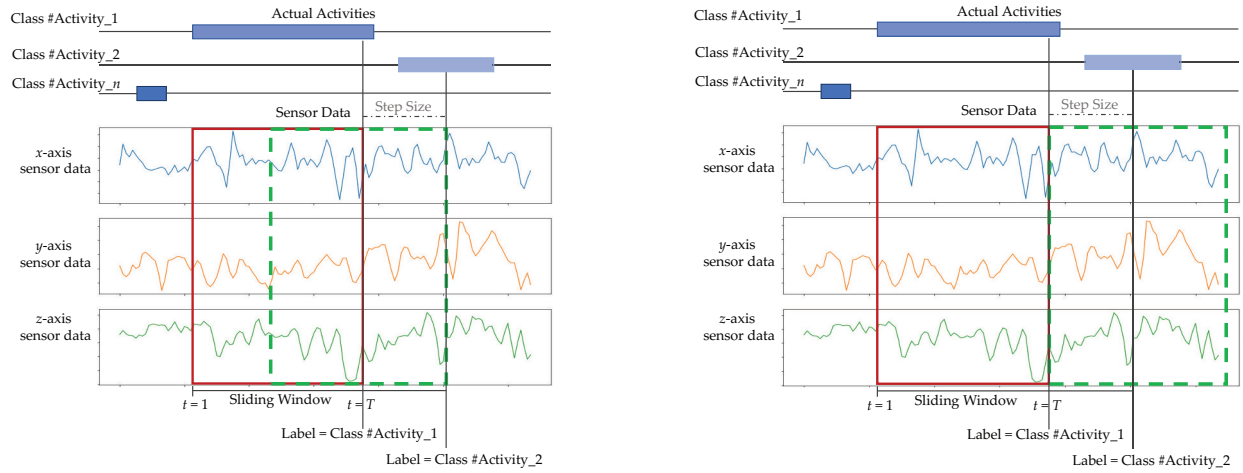
This research examines how altering the window size (5, 10, 20, 30, and 40 seconds) affects training diverse deep learning issues throughout various scenarios. Furthermore, we acknowledge that the recognition ability improves as the window size rises, especially for intricate actions, as seen in Table II and Table III.

Table II shows performance metrics for CNN, LSTM, BiLSTM, and CNN-LSTM models trained on non-overlapping data segments of 5 to 40 seconds. Key findings:

- CNN-LSTM outperforms other models across all window sizes.
- LSTM and BiLSTM performance declines with larger windows (20+ seconds).
- CNN maintains stable performance across window sizes.
- CNN-LSTM achieves highest accuracy (88.09%) at 5 seconds, but remains robust (¿86%) for larger windows.
- Precision, recall, and F1-score trends mirror accuracy.

Results emphasize the importance of window size and architecture selection in sensor-based activity recognition, with CNN-LSTM emerging as the most effective and robust approach.

Table III shows performance metrics for CNN, LSTM, BiLSTM, and CNN-LSTM models trained on 50% overlapping data segments of 5 to 40 seconds. Key findings:



(a) Overlapping temporal window (OW) with 50% overlap

(b) Non-overlapping temporal window (NOW)

Fig. 3: Fixed-width sliding window segmentation strategies: (a) overlapping temporal window (OW) with 50% overlap, and (b) non-overlapping temporal window (NOW) – in both strategies, X, Y, and Z denote the three components of the tri-axial accelerometer sensor data.

TABLE II: Performance metrics of the deep learning models (CNN, LSTM, BiLSTM, and CNN-LSTM) trained and tested with different window sizes (5, 10, 20, 30, and 40 seconds) using the non-overlapping temporal window (NOW) segmentation protocol

Window Sizes	Model	Accuracy	Precision	Recall	F1-score
5 s	CNN	86.37%	82.94%	83.52%	83.12%
	LSTM	85.90%	85.56%	83.59%	84.50%
	BiLSTM	86.59%	86.22%	83.26%	84.64%
	CNN-LSTM	88.09%	85.40%	86.41%	85.83%
10 s	CNN	85.53%	82.20%	85.43%	83.55%
	LSTM	81.31%	81.01%	74.32%	77.21%
	BiLSTM	86.50%	86.46%	84.71%	85.40%
	CNN-LSTM	86.50%	84.41%	85.93%	85.07%
20 s	CNN	82.57%	82.12%	84.17%	82.89%
	LSTM	73.30%	72.51%	66.45%	68.23%
	BiLSTM	81.16%	81.16%	77.76%	78.95%
	CNN-LSTM	87.18%	88.56%	87.46%	87.82%
30 s	CNN	83.24%	85.47%	90.17%	87.44%
	LSTM	64.01%	72.68%	52.72%	59.29%
	BiLSTM	71.20%	74.89%	67.64%	69.98%
	CNN-LSTM	86.08%	87.79%	91.42%	89.24%
40 s	CNN	83.21%	82.87%	87.02%	84.65%
	LSTM	62.65%	75.25%	65.66%	68.91%
	BiLSTM	69.97%	68.88%	72.03%	68.68%
	CNN-LSTM	86.41%	87.19%	90.71%	88.56%

TABLE III: Performance metrics of the deep learning models (CNN, LSTM, BiLSTM, and CNN-LSTM) trained and tested with different window sizes (5, 10, 20, 30, and 40 seconds) using the overlapping temporal window (OW) segmentation protocol with 50% overlapping proportion

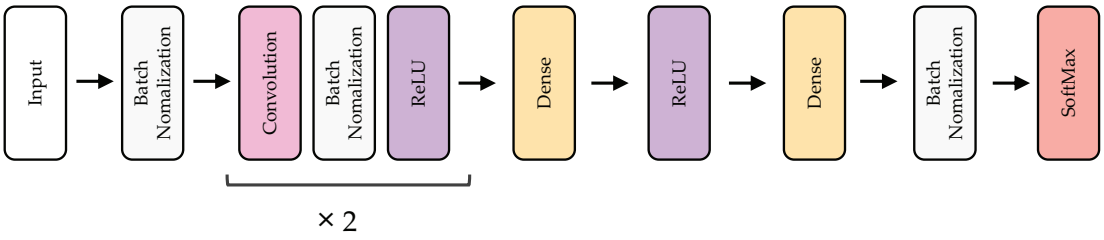
Window Sizes	Model	Accuracy	Precision	Recall	F1-score
5 s	CNN	89.21%	87.26%	88.19%	87.66%
	LSTM	88.79%	86.43%	86.89%	86.59%
	BiLSTM	90.98%	89.00%	90.30%	89.57%
	CNN-LSTM	91.55%	88.21%	90.85%	89.44%
10 s	CNN	87.94%	86.61%	85.41%	85.86%
	LSTM	87.40%	87.99%	85.64%	86.68%
	BiLSTM	88.87%	87.74%	86.48%	87.05%
	CNN-LSTM	91.91%	89.25%	90.93%	90.06%
20 s	CNN	86.38%	86.30%	89.49%	87.77%
	LSTM	79.23%	77.50%	75.13%	75.86%
	BiLSTM	84.43%	85.04%	81.14%	82.83%
	CNN-LSTM	92.11%	91.40%	90.49%	90.89%
30 s	CNN	86.48%	86.29%	89.18%	87.58%
	LSTM	70.39%	76.31%	65.26%	69.79%
	BiLSTM	77.93%	77.96%	72.54%	74.53%
	CNN-LSTM	90.91%	93.08%	93.57%	93.21%
40 s	CNN	85.76%	83.47%	91.61%	87.24%
	LSTM	59.36%	66.10%	49.27%	54.37%
	BiLSTM	71.69%	75.67%	70.80%	72.63%
	CNN-LSTM	91.31%	93.25%	95.44%	94.28%

- CNN-LSTM consistently outperforms other models across all window sizes.
- Overlapping protocol improves performance for all models compared to non-overlapping.
- LSTM performance declines sharply with more oversized windows (30+ seconds).
- BiLSTM and CNN maintain relatively stable performance across window sizes.
- CNN-LSTM achieves highest accuracy (92.11%) at 20 seconds.

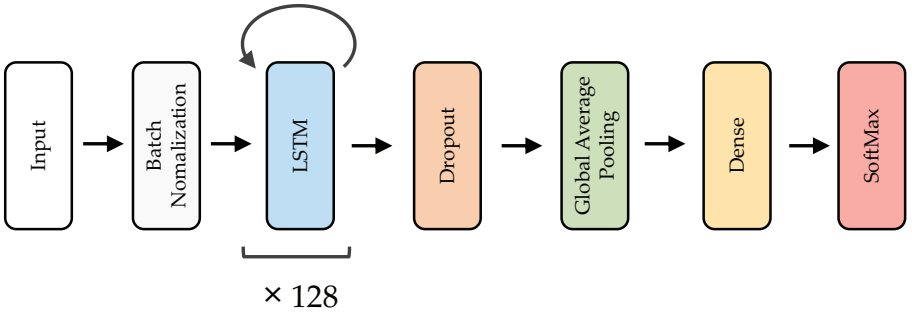
Results confirm CNN-LSTM's superiority for sensor-based activity recognition, especially with overlapping windows. Smaller window sizes generally lead to better performance. The overlapping protocol improves results by capturing more contextual information. Window size selection should consider sensor data characteristics and task requirements.

V. Conclusion and Future Works

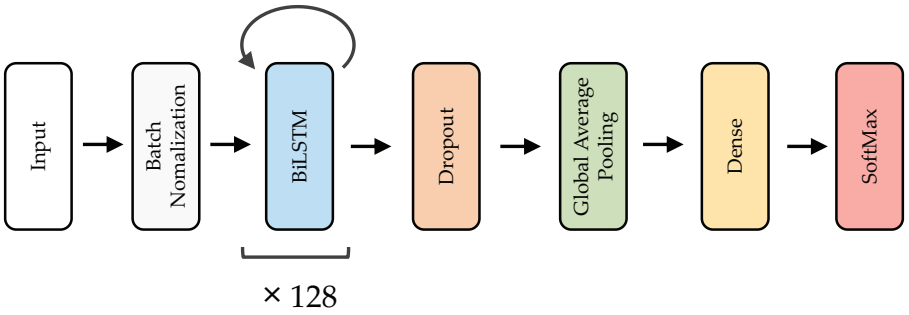
This study investigates how sliding window sizes affect deep learning models' performance in recognizing human activities



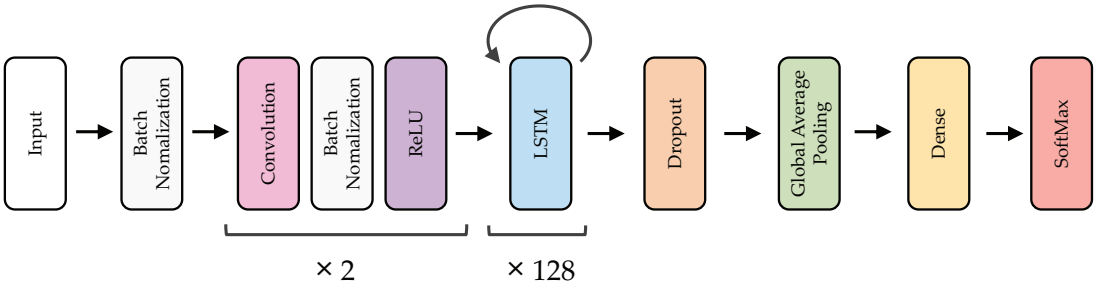
(a) CNN architecture



(b) LSTM architecture



(c) BiLSTM architecture



(d) CNN-LSTM architecture

Fig. 4: Deep learning architectures used in this study.

from smartwatch sensor data. Four models (CNN, LSTM, BiLSTM, CNN-LSTM) were tested on the DHA dataset using 5–40 second windows with overlapping and non-overlapping segmentation. Key findings:

- Smaller windows generally improved model performance.
- CNN-LSTM consistently outperformed other models.
- LSTM and BiLSTM were more sensitive to larger window sizes than CNN.
- Overlapping segmentation enhanced overall performance.

The CNN-LSTM hybrid model proved most effective and robust. Window size selection should consider sensor data characteristics, activity complexity, and application requirements.

Future work could explore different sensor modalities and fusion techniques, advanced architectures for complex dependencies, data augmentation for improved robustness, more extensive, more diverse datasets, and personalized activity recognition models.

ACKNOWLEDGMENT

This research project was supported by University of Phayao (Grant no. FF67-UoE-214); Thailand Science Research and Innovation Fund (Fundamental Fund 2024); National Science, Research and Innovation Fund (NSRF); and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-67-B-10.

REFERENCES

[1] F. Demrozi, C. Turetta, P. H. Kindt, F. Chiarani, R. A. Bacchin, N. Valè, F. Pascucci, P. Cesari, N. Smania, S. Tamburin, and G. Pravadelli, "A low-cost wireless body area network for human activity recognition in healthy life and medical applications," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 4, pp. 839–850, 2023.

[2] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Improving lower limb activity recognition based on inertial sensors using cnn-lstm network," in *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2023, pp. 170–173.

[3] P. Khan, Y. Kumar, and S. Kumar, "Capslstm-based human activity recognition for smart healthcare with scarce labeled data," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 1, pp. 707–716, 2024.

[4] S. Mekruksavanich and A. Jitpattanakul, "Deep residual neural network for aggressive physical activity recognition using surface electromyography sensors," in *2023 IEEE 14th International Conference on Software Engineering and Service Science (ICSESS)*, 2023, pp. 175–178.

[5] J.-S. Kim, "Dnn-based human activity recognition by learning initial motion data for virtual multi-sports," in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, 2021, pp. 373–375.

[6] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Recognizing and understanding sport activities based on wearable sensor signals using deep residual network," in *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2023, pp. 166–169.

[7] J. Wei, B. Yu, H. Zhang, and J. Liu, "Skeleton based graph convolutional network method for action recognition in sports: A review," in *2023 8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*, 2023, pp. 66–70.

[8] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "A hybrid deep learning neural network for recognizing exercise activity using inertial sensor and motion capture system," in *2023 4th International Conference on Big Data Analytics and Practices (IBDAP)*, 2023, pp. 1–5.

[9] S. Mekruksavanich and A. Jitpattanakul, "A lightweight deep residual network for recognizing activities in daily living using channel state information," in *2023 IEEE 14th International Conference on Software Engineering and Service Science (ICSESS)*, 2023, pp. 171–174.

[10] S. B. Rekha and M. V. Rao, "Methodical activity recognition and monitoring of a person through smart phone and wireless sensors," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, 2017, pp. 1456–1459.

[11] S. Kalimuthu, T. Perumal, R. Yaakob, E. Marlisah, and L. Babangida, "Human activity recognition based on smart home environment and their applications, challenges," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, pp. 815–819.

[12] S. Mekruksavanich and A. Jitpattanakul, "Classifying activities of electrical line workers based on deep learning approaches using wrist-worn sensor," in *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2023, pp. 270–274.

[13] N. Hnoohom, A. Jitpattanakul, P. Inluergsri, P. Wongbudsri, and W. Ployput, "Multi-sensor-based fall detection and activity daily living classification by using ensemble learning," in *2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON)*, 2018, pp. 111–115.

[14] W. Sanpote, P. Jantawong, N. Hnoohom, A. Jitpattanakul, and S. Mekruksavanich, "Deep learning approaches for recognizing daily human activities using smart home sensors," in *2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2023, pp. 469–473.

[15] S. Abbas, G. A. Sampedro, S. Alsubai, S. Ojo, A. S. Almadhor, A. A. Hejaili, and L. Strazovska, "Advancing healthcare and elderly activity recognition: Active machine and deep learning for fine- grained heterogeneity activity recognition," *IEEE Access*, vol. 12, pp. 44 949–44 959, 2024.

[16] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Deep learning approaches for har of daily living activities using imu sensors in smart glasses," in *2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2023, pp. 474–478.

[17] D. Noh, H. Yoon, and D. Lee, "A decade of progress in human motion recognition: A comprehensive survey from 2010 to 2020," *IEEE Access*, vol. 12, pp. 5684–5707, 2024.

[18] L. Sun, X. Yang, and C. Hu, "Dswhar: A dynamic sliding window based human activity recognition method," in *2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, 2022, pp. 1421–1426.

[19] M. H. M. Noor, Z. Salcic, and K. I.-K. Wang, "Dynamic sliding window method for physical activity recognition using a single tri-axial accelerometer," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 102–107.

[20] J. Wan, M. J. O'Grady, and G. M. P. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 287–301, Feb 2015.

[21] G. Wang, Q. Li, L. Wang, W. Wang, M. Wu, and T. Liu, "Impact of sliding window length in indoor human motion modes and pose pattern recognition based on smartphone sensors," *Sensors*, vol. 18, no. 6, 2018.

[22] S. Mekruksavanich and A. Jitpattanakul, "Rnn-based deep learning for physical activity recognition using smartwatch sensors: A case study of simple and complex activity recognition," *Mathematical Biosciences and Engineering*, vol. 19, no. 6, pp. 5671–5698, 2022.

[23] J.-H. Wang, J.-J. Ding, Y. Chen, and H.-H. Chen, "Real time accelerometer-based gait recognition using adaptive windowed wavelet transforms," in *2012 IEEE Asia Pacific Conference on Circuits and Systems*, 2012, pp. 591–594.

[24] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous Intelligence and Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 548–562.

[25] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, "Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis," in *2010 5th International Conference on Future Information Technology*, 2010, pp. 1–6.

[26] Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer," in *Hybrid Artificial Intelligent Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 460–467.

[27] P. Siirtola and J. Röning, "User-independent human activity recognition using a mobile phone: Offline recognition vs. real-time on device recognition," in *Distributed Computing and Artificial Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 617–627.

[28] N. Mairittha, T. Mairittha, and S. Inoue, "On-device deep personalization for robust activity data collection," *Sensors*, vol. 21, no. 1, 2021.

[29] M. Ebner, T. Fetzer, M. Bullmann, F. Deinzer, and M. Grzegorzek, "Recognition of typical locomotion activities based on the sensor data of a smartphone in pocket or hand," *Sensors*, vol. 20, no. 22, 2020.

[30] M.-C. Kwon and S. Choi, "Recognition of daily human activity using an artificial neural network and smartwatch," *Wireless Communications and Mobile Computing*, vol. 2018, no. 1, p. 2618045, 2018.

turnitin Page 180 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Integrating In-Ear Wearable Sensors with Deep Learning for Head and Facial Movement Analysis

Sakorn Mekruksavanich[1], Ponnipa Jantawong[1] and Anuchit Jitpattanakul[2,*]
[1]*Department of Computer Engineering, School of Information and Communication Technology*
University of Phayao, Phayao, Thailand
sakorn.me@up.ac.th and ponnipa.jantawong@gmail.com
[2]*Intelligent and Nonlinear Dynamic Innovations Research Center, Department of Mathematics*
Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
anuchit.j@sci.kmutnb.ac.th

***Abstract*—Accurate and inconspicuous identification of eating behavior has substantial implications for wellness surveillance, nutritional tracking, and technological assistance. Conventional approaches for detecting eating habits frequently rely on invasive or unfeasible techniques, which restrict their practicality in real-life situations. This research introduces an innovative eating recognition system that utilizes head and face motions recorded by in-ear wearing sensors and advanced deep learning algorithms. Our approach is a non-invasive and simple-to-use method that utilizes the unique patterns of head and face motions related to eating actions. We use a specially developed in-ear wearable sensor from the EarSet dataset to gather movement information while individuals are eating. The dataset comprises a range of cranial and facial motions recorded from 30 individuals. The unprocessed sensor data is subjected to pre-processing to eliminate interference and extract pertinent characteristics. We use advanced deep learning models, such as CNN, LSTM, BiLSTM, GRU, BiGRU, and our novel hybrid model CNN-LSTM, to acquire knowledge of distinctive patterns and identify eating occurrences. Our eating recognition system has been extensively tested and proven successful. The CNN-LSTM model achieved an accuracy of 95.12% and an F1-score of 95.09% during the 5-fold cross-validation studies. Our technique demonstrates superior efficiency and applicability compared to standard deep learning algorithms, as a comparative study shows. The suggested system can combine in-ear wearable sensors with deep learning to provide inconspicuous and accurate identification of eating activities. This study enhances the development of eating detection technology and introduces new opportunities for individualized wellness tracking and helpful applications. Subsequent research will prioritize the enlargement of the dataset, integration of supplementary modalities, enhancement of deep learning models for equipment with limited resources, and assessment of the long-term practicality and user reception of in-ear wearable sensors for detecting eating activities.**

***Keywords*—eating detection, in-ear wearable sensors, deep learning, head movements, facial movements, health monitoring**

I. Introduction

The identification of dietary intake has received considerable interest in the last few decades owing to its possible uses for wellness tracking, nutritional following, and assistive technology [1]. Precise and inconspicuous surveillance of eating behaviors could offer significant observations about a person's eating patterns, allowing customized therapies and fostering better ways of living [2]. Conventional methods for detecting eating habits usually include the use of periodic food journaling or mobile devices that record audio or electromyography (EMG) data [3], [4]. Nevertheless, these approaches may be invasive and onerous, increasing privacy apprehensions.

Improvements in wearable sensor technology and deep learning have recently emerged, enabling the identification of eating behaviors employing non-invasive and user-friendly methods [5]. Specifically, inertial measurement unit (IMU) sensors, which include accelerometers and gyroscopes, have demonstrated potential in recording the unique movement characteristics related to eating actions [6]–[8]. These sensors are often encountered in wearable devices and may be readily incorporated into daily routines [9]–[11].

Conventional machine learning methods have been widely studied to detect eating behavior through wearable IMU sensors. These approaches need a two-step procedure consisting of feature extraction and classification. During the feature extraction, manually designed features are obtained from the unprocessed sensor data to acquire pertinent details on eating behaviors. Typical characteristics include statistical measurements such as mean, variance, and entropy, time-domain features like zero-crossing rate and root mean square, and frequency-domain information such as Fourier coefficients and wavelet transformations [12]. The collected characteristics are further inputted into conventional machine learning methods, such as support vector machines, decision trees, and random forests, for categorization [13]. Although these conventional methods have shown particular effectiveness in detecting eating behavior, their performance primarily depends on the accuracy of manually designed features, which can sometimes fail to represent the intricate structures present in sensor data. Furthermore, the feature engineering process may be laborious and demands specialized knowledge in a particular field, constraining the scalability and applicability of these approaches.

Convolutional neural networks (CNNs), a kind of deep learning approach, have shown outstanding effectiveness in recognizing pattern operations, such as identifying human

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

activities [14]–[16]. CNNs can acquire and extract significant characteristics from unprocessed sensor data autonomously, hence obviating the need for manually constructed feature engineering. This renders them very suitable for evaluating feeding behaviors' intricate and nuanced movement patterns.

This study introduces a new eating detection approach that utilizes head and face motions recorded by in-ear IMU sensors and an integrated deep learning structure known as CNN-LSTM. Our objective is to create a reliable and accurate approach to detecting eating behavior using IMU data. We intend to do this by employing CNN-LSTM and ensuring that the system can be readily used on wearable devices with limited resources. This study's primary contributions are as follows:

- Our study introduces an innovative eating detection system integrating in-ear IMU sensors with the CNN-LSTM deep learning structure.
- We meticulously evaluate the suggested approach using a substantial dataset, demonstrating its superior efficiency when compared to current methods and instilling confidence in its reliability.
- We explore our eating-detecting method's possible uses and potential applications.

II. Related Works

This section provides an overview of the current research on identifying eating behaviors and using advanced machine-learning methods to analyze data collected from sensors. The relevant research is categorized into two subsections: (1) methods for detecting eating behavior and (2) deep learning techniques for analyzing sensor data.

A. Eating Detection Approaches

The field of eating detection has seen significant research activity, with several techniques presented in the past few decades. Conventional approaches often depend on manually recording food intake or relying on self-reporting, which may be onerous and susceptible to mistakes [17]. To address these constraints, scientists have investigated the utilization of wearable sensors for autonomous eating recognition [18], [19].

An established method involves using audio sensors to record noises related to eating actions, including chewing and swallowing [20], [21]. These technologies often use signal processing and machine learning algorithms to categorize eating occurrences based on their auditory characteristics. Nevertheless, audio-based methods may be susceptible to disruptions caused by ambient noise and may also give rise to privacy issues.

Another study area uses EMG sensors to identify muscle contractions associated with eating. EMG sensors detect the muscles' electrical signals near the mouth and jaw, offering a more explicit representation of eating behaviors [3], [22]. Nevertheless, EMG sensors can be obtrusive and may induce pain with extended periods of use.

In recent times, scientists have investigated the utilization of movement sensors, including accelerometers and gyroscopes, to detect eating behaviors [6]. These sensors are capable of detecting the unique head and hand motions related to eating actions. Motion-based techniques are less invasive and more protective of privacy than audio and EMG-based methods.

Although there have been advancements in detecting eating behavior, current methods generally need more precision, especially in real-life situations. Moreover, several techniques depend on manually designed characteristics and conventional machine learning algorithms, which could not accurately represent the intricate patterns present in sensor data.

B. Deep Learning Methods

The subject of sensor data processing has been transformed by deep learning methods, namely CNNs. CNNs can autonomously acquire hierarchical representations from unprocessed sensor input, therefore obviating human feature engineering requirements [23], [24].

Within human activity detection, CNNs have been extensively used to assess data collected from a range of sensors, including accelerometers, gyroscopes, and EMG sensors [25]. This research has shown that CNNs help preserve the spatial and temporal relationships in sensor data, resulting in improved accuracy in classification compared to standard machine learning methods.

In addition, deep learning models, including recurrent neural networks (RNNs) and long short-term memory (LSTM) networks [26]–[29], have been particularly developed to process time series data. These designs can represent the time-dependent changes in sensor data, which makes them very suitable for assessing eating behaviors that display temporal patterns.

Our objective in this study is to close this divide by introducing a new system for detecting eating habits. This system utilizes in-ear IMU sensors with a hybrid deep-learning framework. We aim to create a precise and user-friendly eating identification system using the powerful hybrid deep learning model and discreet in-ear sensors.

III. The Proposed Framework

The eating identification architecture utilized in this study consists of four primary processes: data gathering, data preprocessing, data production, model training, and assessment, as seen in Fig. 1.

A. EarSet Dataset

The EarSet dataset [30] is a comprehensive collection designed to study how body, head, and face motions affect the shape of the Photoplethysmography (PPG) wave recorded at the ear and the accurate assessment of vital signs. This dataset was created using a specialized and adaptable research platform featuring a unique ear-tip design. This design incorporates a 3-channel PPG (green, red, infrared) and a 6-axis motion sensor (IMU) consisting of an accelerometer and gyroscope, all positioned within the same ear-tip. This setup allows for the simultaneous collection of PPG data from

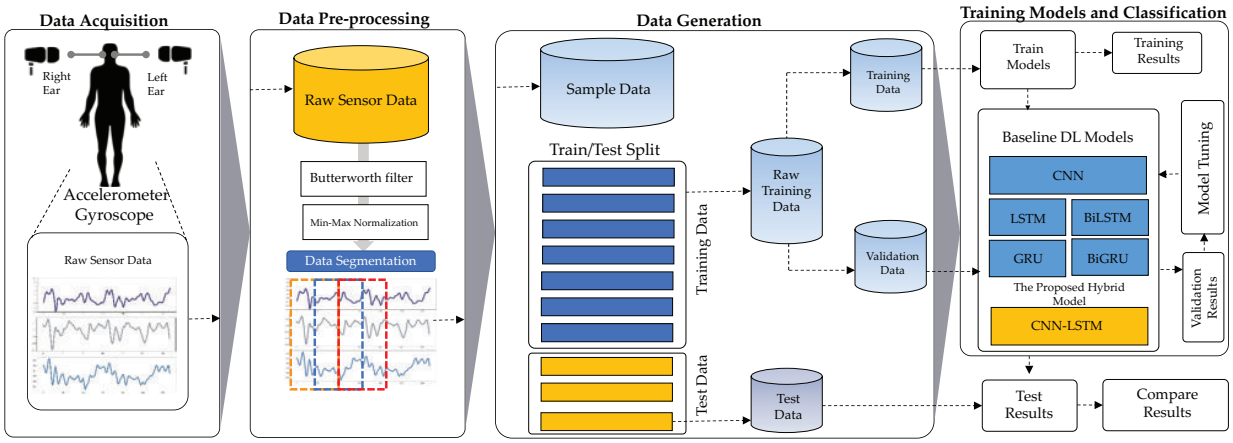


Fig. 1: Overview of the proposed eating detection framework using IMU sensors from in-ear wearable devices.

different spatial locations and wavelengths, capturing motion patterns from both ears, resulting in 18 data streams.

The dataset includes information on 16 distinct head and face movements, categorized as one-time or continuous actions. One-time motions include:

- head movements (nodding, shaking, tilting),
- eye movements (vertical and horizontal, brow-raising, brow-lowering, winking), and
- mouth movements (lip pulling, chin raising, mouth stretching).

Continuous movements cover everyday actions like chewing and speaking. Additionally, the dataset collects motion and PPG data during various full-body activities of varying intensities, such as walking and running. Other vital signs, including heart rate, heart rate variability, breathing rate, and raw ECG, were measured using a high-quality medical chest device.

The dataset, a rich resource, comprises over 17 hours of data from 30 participants representing diverse gender and race, with an average age of 28.9 years (SD = 6.11). This extensive dataset empowers researchers to analyze in-ear PPG signals in relation to movement, device placement (left or right ear), and different configurations, balancing data quality and power usage.

This study is dedicated to leveraging the motion data obtained from the 6-axis IMU sensor within the EarSet's innovative ear-tip design. The IMU sensor, comprising an accelerometer and gyroscope, provides a comprehensive 6 degrees-of-freedom motion profile of head and face movements. By utilizing only IMU data, the study aims to efficiently identify and categorize eating behaviors, without processing PPG signals, thereby enhancing energy and computational efficiency and protecting privacy by avoiding the collection of sensitive physiological data. The ultimate goal is to develop a highly accurate, non-intrusive system for detecting eating activities based on unique head and face movement patterns recorded by the IMU sensor.

B. Data Pre-processing

The unprocessed sensor data underwent a sequence of pre-processing procedures to guarantee data integrity and prepare it for additional examination. Initially, noise elimination methods were used to mitigate the influence of undesirable artifacts and enhance the signal-to-noise ratio. The data was first processed using a median filter to remove unusual cases and get a smoother result. Then, a third-order low-pass Butterworth filter with a cutoff frequency of 20 Hz was used to eliminate high-frequency interference while retaining the critical movement data.

Subsequently, the process of data normalization was executed employing the Min-Max approach, a significant step that standardizes the sensor data within a uniform range. This process serves to alleviate the impact of inter-subject variability and guarantees the comparability of data collected from different individuals and sensors. The Min-Max normalization function assigns a numerical value to each characteristic within a range of 0 to 1, utilizing the following mathematical expression:

$$X_{normalized} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (1)$$

Let X represent the original value, X_{min} represent the lowest result of the feature, X_{max} represent the highest result of the feature, and $X_{normalized}$ represent the normalized result.

Following normalization, the pre-processed sensor data was segmented using a fixed-width sliding window technique. A window duration of 2 seconds with a 50% overlap ratio was chosen to capture the average length of eating movements and provide sufficient temporal context for precise categorization. This sliding window approach balances the collection of immediate and prolonged eating habits, optimizing accuracy and computational efficiency.

To address the imbalance between eating and non-eating behaviors, we employed the synthetic minority over-sampling

technique (SMOTE) [31]. SMOTE generates artificial instances of the minority class (eating activity) by interpolating between existing instances in the feature space. This strategy balances the dataset and reduces bias towards the dominant class during model training. The augmented dataset, now with equal eating and non-eating samples, was used for feature extraction and model construction.

Our objective is to improve the accuracy and representativeness of the IMU data by employing pre-processing techniques such as noise reduction, data normalization, segmentation, and data augmentation. This will enable us to achieve greater accuracy and dependability in eating identification by employing deep learning methods.

C. The Proposed CNN-LSTM Model

CNNs are highly effective for extracting features from one-dimensional sequence data, like multivariate time series. They can also be used hybrid by combining them with an LSTM backend. In this configuration, the CNN interprets input sub-sequences, which are then sequentially processed by the LSTM for further analysis. This hybrid model, the CNN-LSTM model, leverages CNN layers to extract features from the input data while the LSTM component handles sequence prediction. The CNN-LSTM model analyzes sub-sequences from the main sequence as blocks, with the CNN extracting key features from each block and the LSTM interpreting these features. Fig. 2 depicts the CNN-LSTM architecture, and Table I lists the associated hyperparameters.

IV. Experiments and Results

All experiments in this study are performed on the Google Colab Pro+ platform with a Tesla V100 GPU. The experiments use Python 3.6.9 and TensorFlow 2.2.0, Keras 2.3.1, Scikit-Learn, Numpy 1.18.5, and Pandas 1.0.5 libraries. This research evaluates the detection abilities of deep learning models constructed with CNN and RNN architectures. IMU sensor data over a 2-second period is utilized for each test to train and evaluate these models.

The EarSet dataset is filtered to include only sensor data about eating and drinking activities. This data is processed using a 5-fold cross-validation method. Numerous experiments are conducted to examine the detection capabilities of five standard deep learning models and a new hybrid deep learning model. The detection interpretation of each model is presented in Table II.

The results in Table II illustrate the detection performance of five fundamental deep learning models (CNN, LSTM, BiLSTM, GRU, and BiGRU) and a novel hybrid model (CNN-LSTM) for identifying eating activities using IMU sensor data from the EarSet dataset.

The BiGRU model, a standout among the baseline models, achieved the highest accuracy with an impressive 92.41% (±6.12%) and an F1-score of 92.37% (±6.18%). This robust performance underscores the effectiveness of the bidirectional GRU architecture in capturing sequential patterns in the IMU data, leading to accurate eating detection. The GRU model also demonstrated strong performance, recording an accuracy of 85.27% (±13.90%) and an F1-score of 84.89% (±14.50%).

The CNN model, designed to extract spatial features from the IMU data, demonstrated its effectiveness with an accuracy of 85.80% (±7.34%) and an F1-score of 85.45% (±7.81%). This robust performance reassures the audience that the convolutional layers can effectively learn distinctive patterns in the motion data associated with eating behaviors.

However, the LSTM and BiLSTM models that captured long-term dependencies performed worse than the other baseline models. The LSTM model reached an accuracy of 66.48% (±14.63%) and an F1-score of 61.17% (±20.21%). The BiLSTM model achieved an accuracy of 58.14% (±7.25%) and an F1-score of 51.25% (±10.41%). These results suggest that LSTM-based models may need support to learn eating-related patterns from IMU data effectively.

The CNN-LSTM model, which combines convolutional and recurrent layers, outperformed all baseline models. It achieved an impressive accuracy of 95.12% (±1.56%) and an F1-score of 95.09% (±1.59%). The hybrid design, with CNN layers extracting important spatial features and the LSTM layer capturing temporal relationships, likely contributes to its superior performance. This combination allows the CNN-LSTM model to learn and recognize eating-related patterns in the motion data effectively.

Moreover, the CNN-LSTM model exhibited much lower performance metrics standard deviations than the baseline models. This indicates greater consistency and stability across multiple cross-validation folds, highlighting its robustness and generalization ability.

V. Conclusion

This research introduces an innovative eating detection system that utilizes head and face movements captured by in-ear IMU sensors, leveraging a hybrid deep learning framework called CNN-LSTM. Using motion data from the EarSet dataset, we have developed an accurate and convenient method for non-invasive eating detection.

Our approach incorporates data preprocessing techniques such as noise reduction, normalization, segmentation, and augmentation to improve the quality and representativeness of the IMU data. We assessed the performance of our CNN-LSTM model against five other deep learning models using a 5-fold cross-validation approach.

The experimental results confirm the effectiveness of our system. The CNN-LSTM model achieved an accuracy of 95.12% and an F1-score of 95.09%. Compared to standard models, this superior performance underscores the potential of combining in-ear IMU sensors with hybrid deep-learning architectures for detecting eating activities.

The proposed method has significant health monitoring, food tracking, and assistive technology applications. Our technique offers precise and non-intrusive detection of eating behavior, which can support personalized therapies, promote healthy habits, and assist individuals with eating disorders or other dietary-related conditions.

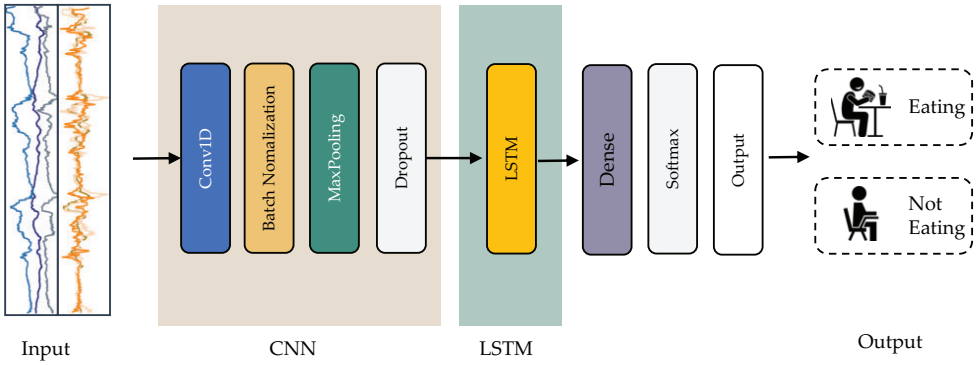


Fig. 2: The architecture of CNN-LSTM.

TABLE I: The summary of hyperparameters for CNN-LSTM networks proposed in this work

Stage	Hyperparameters		Values
Architecture	**CNN Block**		
	1D-Convolution	Kernel Size	5
		Stride	1
		Filters	256
	Batch Normalization		-
	Activation		ReLU
	Max Pooling		2
	Dropout		0.25
	LSTM Block		
	LSTM	Neural	128
	Dense		128
	Activation		SoftMax
Training	Loss Function		Cross-entropy
	Optimizer		Adam
	Batch Size		64
	Number of Epochs		200

TABLE II: Performance metrics of deep learning models using IMU data from in-ear wearable device

Model	Recognition Effectiveness		
	Accuracy	Loss	F1-score
CNN	85.80%(±7.34%)	0.31(±0.11)	85.45%(±7.81%)
LSTM	66.48%(±14.63%)	0.54(±0.18)	61.17%(±20.21%)
BiLSTM	58.14%(±7.25%)	0.64(±0.07)	51.25%(±10.41%)
GRU	85.27%(±13.90%)	0.29(±0.21)	84.89%(±14.50%)
BiGRU	92.41%(±6.12%)	0.17(±0.11)	92.37%(±6.18%)
CNN-LSTM	95.12%(±1.56%)	0.14(±0.04)	95.09%(±1.59%)

ACKNOWLEDGMENT

This research project was supported by University of Phayao (Grant no. FF67-UoE-214); Thailand Science Research and Innovation Fund (Fundamental Fund 2024); National Science, Research and Innovation Fund (NSRF); and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-67-B-10.

REFERENCES

[1] M. A. Subhi, S. H. Ali, and M. A. Mohammed, "Vision-based approaches for automatic food recognition and dietary assessment: A survey," *IEEE Access*, vol. 7, pp. 35 370–35 381, 2019.

[2] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Smartwatch-based eating detection and cutlery classification using a deep residual network with squeeze-and-excitation module," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, 2022, pp. 301–304.

[3] G. Idris, C. Smith, B. Galland, R. Taylor, C. J. Robertson, and M. Farella, "Home-based monitoring of eating in adolescents: A pilot study," *Nutrients*, vol. 13, no. 12, 2021.

[4] R. Zhang and O. Amft, "Retrieval and timing performance of chewing-based eating event detection in wearable sensors," *Sensors*, vol. 20, no. 2, 2020.

[5] W. Bangamuarachchi, A. Chamantha, L. Meegahapola, S. Ruiz-Correa, I. Perera, and D. Gatica-Perez, "Sensing eating events in context: A smartphone-only approach," *IEEE Access*, vol. 10, pp. 61 249–61 264, 2022.

[6] S. Mekruksavanich, P. Jantawong, N. Hnoohom, and A. Jitpattanakul,

"Deep learning networks for eating and drinking recognition based on smartwatch sensors," in *2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2022, pp. 106–111.

[7] C. Wang, T. S. Kumar, W. De Raedt, G. Camps, H. Hallez, and B. Vanrumste, "Eating speed measurement using wrist-worn imu sensors towards free-living environments," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–12, 2024.

[8] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Smartwatch-based eating detection and cutlery classification using a deep residual network with squeeze-and-excitation module," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, 2022, pp. 301–304.

[9] N. Hnoohom, N. Maitrichit, S. Mekruksavanich, and A. Jitpattanakul, "Deep learning approaches for unobtrusive human activity recognition using insole-based and smartwatch sensors," in *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, 2022, pp. 1–5.

[10] A. Jitpattanakul and S. Mekruksavanich, "Enhancing sensor-based human activity recognition using efficient channel attention," in *2023 IEEE SENSORS*, 2023, pp. 1–4.

[11] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Human activity recognition and payload classification for low-back exoskeletons using deep residual network," in *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2023, pp. 313–317.

[12] J.-H. Li, P.-W. Yu, H.-C. Wang, C.-Y. Lin, Y.-C. Lin, C.-P. Liu, C.-Y. Hsieh, and C.-T. Chan, "Multi-sensor fusion approach to drinking activity identification for improving fluid intake monitoring," *Applied Sciences*, vol. 14, no. 11, 2024.

[13] A. I. Alexan, A. R. Alexan, and S. Oniga, "Real-time machine learning for human activities recognition based on wrist-worn wearable devices," *Applied Sciences*, vol. 14, no. 1, 2024.

[14] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Accuracy improvement of complex sensor-based activity recognition using hybrid cnn," in *2022 6th International Conference on Information Technology (InCIT)*, 2022, pp. 454–457.

[15] H. Madokoro, S. Nix, H. Woo, and K. Sato, "A mini-survey and feasibility study of deep-learning-based human activity recognition from slight feature signals obtained using privacy-aware environmental sensors," *Applied Sciences*, vol. 11, no. 24, 2021.

[16] N. Hnoohom, N. Maitrichit, S. Mekruksavanich, and A. Jitpattanakul, "Hierarchical human activity recognition based on smartwatch sensors using branch convolutional neural networks," in *Multi-disciplinary Trends in Artificial Intelligence*. Cham: Springer International Publishing, 2022, pp. 52–60.

[17] Z. Tang, A. Patyk, J. Jolly, S. P. Goldstein, J. G. Thomas, and A. Hoover, "Detecting eating episodes from wrist motion using daily pattern analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 2, pp. 1054–1065, 2024.

[18] R. Zhang and O. Amft, "Monitoring chewing and eating in free-living using smart eyeglasses," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 1, pp. 23–32, 2018.

[19] A. Saphala, R. Zhang, T. N. Thái, and O. Amft, "Non-contact temporalis muscle monitoring to detect eating in free-living using smart eyeglasses," in *2022 IEEE-EMBS International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2022, pp. 1–4.

[20] S. Päßler and W.-J. Fischer, "Food intake monitoring: Automated chew event detection in chewing sounds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 278–289, 2014.

[21] A. Nakamura, T. Saito, D. Ikeda, K. Ohta, H. Mineno, and M. Nishimura, "Automatic detection of chewing and swallowing using multichannel sound information," in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, 2021, pp. 173–175.

[22] R. Zhang and O. Amft, "Free-living eating event spotting using emg-monitoring eyeglasses," in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2018, pp. 128–132.

[23] G. Ascioglu and Y. Senol, "Activity recognition using different sensor modalities and deep learning," *Applied Sciences*, vol. 13, no. 19, 2023.

[24] S. Mekruksavanich, P. Jantawong, D. Tancharoen, and A. Jitpattanakul, "A convolutional neural network for ultra-wideband radar-based hand gesture recognition," in *2023 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC)*, 2023, pp. 1–4.

[25] J. Gilmore and M. Nasseri, "Human activity recognition algorithm with physiological and inertial signals fusion: Photoplethysmography, electrodermal activity, and accelerometry," *Sensors*, vol. 24, no. 10, 2024.

[26] S. Mekruksavanich and A. Jitpattanakul, "Lstm networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, vol. 21, no. 5, 2021.

[27] M. Z. Uddin and J. Torresen, "Activity recognition using smartphone sensors, robust features, and recurrent neural network," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2019, pp. 1–6.

[28] S. Mekruksavanich, P. Jantawong, N. Hnoohom, and A. Jitpattanakul, "Heterogeneous recognition of human activity with cnn and rnn-based networks using smartphone and smartwatch sensors," in *2022 3rd International Conference on Big Data Analytics and Practices (IBDAP)*, 2022, pp. 21–26.

[29] W.-H. Chen, C. A. Betancourt Baca, and C.-H. Tou, "Lstm-rnns combined with scene information for human activity recognition," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–6.

[30] A. Montanari, A. Ferlini, A. N. Balaji, C. Mascolo, and F. Kawsar, "Earset: A multi-modal dataset for studying the impact of head and facial movements on in-ear ppg signals," *Scientific Data*, vol. 10, no. 1, p. 850, Dec 2023.

[31] D. Elreedy and A. F. Atiya, "A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance," *Information Sciences*, vol. 505, pp. 32–64, 2019.

Ensemble Deep Learning Network for Enhancing Performances of Sensor-based Physical Activity Recognition Based on IMU Sensor Data

Sakorn Mekruksavanich[1], Ponnipa Jantawong[1] and Anuchit Jitpattanakul[2,*]
[1]*Department of Computer Engineering, School of Information and Communication Technology*
University of Phayao, Phayao, Thailand
sakorn.me@up.ac.th and ponnipa.jantawong@gmail.com
[2]*Intelligent and Nonlinear Dynamic Innovations Research Center, Department of Mathematics*
Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
anuchit.j@sci.kmutnb.ac.th

***Abstract*—The application of wearable sensors to identify physical activities has drawn significant attention in the healthcare and fitness industries. This study presents a novel ensemble deep learning network to enhance the accuracy of physical movement recognition utilizing data from inertial measurement unit (IMU) sensors. The proposed Ens-CNN-LSTM method merges a convolutional neural network (CNN) with a long short-term memory (LSTM) network. This integration capitalizes on their strengths in analyzing sequential IMU data. The model further incorporates a random forest classifier to finalize predictions. Our proposed model was evaluated using the PAMAP2 benchmark dataset for human activity recognition. This dataset includes data from multiple individuals performing various activities recorded with IMU sensors. We assessed the ensemble model's performance against individual deep learning models through a 5-fold cross-validation method. Results reveal a significant enhancement in overall accuracy, achieving 99.63%, compared to 96.19% for CNN models and 97.35% for LSTM models. The Ens-CNN-LSTM model demonstrated remarkable performance with an accuracy of 99.61%, a recall of 99.63%, and an F1-score of 99.61%. It surpassed the individual models in all metrics. Furthermore, the ensemble model showed increased robustness to variations among individuals and noise in sensor data. The proposed approach improves the accuracy of physical activity recognition and extends its applicability to other domains and activities.**

***Keywords*—human activity recognition, deep learning, ensemble learning, CNN, LSTM, wearable sensors**

I. Introduction

The utilization of devices with sensors for human activity recognition (HAR) has gained significant importance across several areas, such as medical services [1]–[3], physical fitness [4]–[6], working [7] and interaction between humans and computers [8]. Accurate recognition and categorization of movement patterns could result in substantial advancements in individualized health tracking, evaluation of athletic achievements, and assistive technology [9]. Inertial measurement units (IMUs) have been a favored option among the range of sensing technologies owing to their small size, efficient energy use, and capability to record intricate movement information [10], [11].

The topic of HAR has been transformed by machine learning methods, which provide practical tools for automatically detecting and categorizing various physical behaviors based on sensor data [12]. These methods, such as support vector machines (SVM), random forests, and k-nearest neighbors (k-NN), have been extensively used for HAR applications and have achieved significant success. Their practicality and success in real-world applications instill confidence in their continued use. In general, these approaches entail extracting features from raw sensor data and then classifying them. Nevertheless, these models often need specialized knowledge in order to design features and could encounter difficulties when dealing with intricate, multi-dimensional data.

In the past decade, deep learning approaches have demonstrated outstanding effectiveness in handling and assessing intricate time-series data from IMU sensors [13]–[21]. Convolutional neural networks (CNNs) have been shown to be very efficient at collecting spatial characteristics from sensor data. In contrast, long short-term memory (LSTM) networks achieve excellence in extracting temporal connections. Nevertheless, particular deep learning models often encounter difficulties when dealing with the intrinsic problems of HAR, including inter-subject fluctuation, noise in sensor data, and the intricate nature of individual motions [22].

In order to overcome these restrictions, this research suggests an innovative ensemble deep learning technique that merges the advantages of several deep learning structures. This technique, which is at the forefront of research in the field, incorporates CNNs and LSTMs to better represent both spatial and temporal characteristics of IMU data compared to using these models individually. The innovative nature of this technique is sure to intrigue and interest the audience. This technique is specifically developed to improve the general accuracy and resilience of identifying physical activity in various disciplines and actions. The primary contributions of

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

this work are outlined below:

1) We propose the implementation of an innovative ensemble deep learning framework that merges a CNN with an LSTM model. This pioneering approach has the potential to significantly enhance the effectiveness of HAR, inspiring hope for future advancements in the field.
2) The suggested ensemble approach will be meticulously evaluated using the PAMAP2 benchmark dataset, a substantial resource that encompasses a wide range of physical activities and data from numerous individuals.
3) A comparative evaluation of the ensemble methodology against individual deep learning models is of utmost importance. This approach will ensure that the audience is well-informed and knowledgeable about the research methodology.

II. Related Works

A. Deep Learning for Sensor-based HAR (S-HAR)

Current studies in HAR have shown drawbacks in traditional machine learning methods that impact their capacity to recognize human activities accurately [23]. An important limitation is a reliance on handmade characteristics, which is strongly influenced by the decision-maker's experience and comprehension [24]. Nevertheless, deep learning has quickly become a practical option, efficiently resolving these constraints.

Recently, academics have suggested many deep-learning techniques to address time-series classification difficulties in HAR. These investigations have assessed the identification skills of different learning models by utilizing diverse benchmark activity datasets. CNNs [25], [26] and LSTMs [27] are two main models that have shown their effectiveness in solving HAR issues utilizing smartphone data. These models provide suitable assessment criteria for assessing their performance. This research examines the efficacy of these two models in accurately identifying hand gestures based on data collected from smartwatches.

B. Hybrid Deep Learning for S-HAR

Although CNNs and recurrent neural networks (RNNs) have shown encouraging outcomes in S-HAR applications alone, a new focus has been on hybrid deep learning structures combining both models. These hybrid techniques use the advantages of both CNNs and RNNs to efficiently collect spatial and temporal characteristics from sensor data [28].

The CNN-LSTM model is a prominent hybrid design that combines convolutional layers to extract features and LSTM layers to describe temporal relationships. This method involves using a CNN to identify distinct spatial characteristics from unprocessed sensor input and an LSTM to record extended temporal relationships within the sequence of features [29]. The integration has improved efficiency in several HAR tests compared to separate CNN or LSTM models [30].

Another emerging hybrid structure is the CNN-GRU model, which replaces LSTM layers with gated recurrent units (GRUs). GRUs are RNNs with RNN with fewer parameters than LSTM networks. This reduction in parameters allows GRUs to be more computationally efficient [31]. CNN-GRU models have been shown to be effective in solving S-HAR issues, with improved efficiency and decreased training duration in comparison to CNN-LSTM models [32].

The latest advancement in hybrid deep learning for S-HAR has also investigated the incorporation of attention processes. These processes enable the model to prioritize the most pertinent features or time intervals, improving its ability to detect significant sensor data patterns [33]. An example of a suggested model is the CNN-LSTM with attention, in which the attention mechanism allocates different weights to each time step of the LSTM outcomes depending on their importance [34]. The hybrid model, which utilizes attention mechanisms, has shown enhanced efficacy in accurately identifying intricate human behaviors.

III. The Proposed Methodology

This part presents our methodology for evaluating the importance of hybrid learning approaches in S-HAR. As shown in Fig. 1, our workflow is a modified version of the activity recognition chain and summarizes our approach to recognizing human behavior. Our technique starts with data pre-processing, eliminating extraneous information and standardizing signals from many origins to cleanse the dataset. Subsequently, we use a windowing approach to carry out data segmentation. The generated sample data is then fed into a deep learning network, which produces a set of projected action labels. Ultimately, we assess the effectiveness of our approach by applying established HAR criteria such as accuracy, precision, recall, and F1-score. These measurements enable us to evaluate the efficacy of our hybrid learning strategy in accurately identifying human behaviors based on sensor data. This efficient strategy allows us to methodically examine and contrast the effectiveness of our hybrid learning method with existing techniques in the area of S-HAR.

A. PAMAP2 Dataset

Reiss and Stricker [35] presented the PAMAP2 dataset for tracking movement patterns. It includes captures from nine people, including both male and female participants. The participants were instructed to participate in various lifestyle operations, including leisure activities and domestic responsibilities. These actions included ascending and descending stairs, ironing clothes, vacuum cleaning, laying down, strolling, standing, driving a vehicle, conducting Nordic walks, and others. Residential operations included various movements such as reclining, sitting, standing, going up, and going downstairs.

For more than ten hours, the researchers gathered data on the subjects' physical activity employing three IMUs placed on their dominant wrist, thorax, and ankle. The sampling rate of each IMU was 100 Hz, whereas the heart rate sensor recorded data at a rate of 9 Hz. Furthermore, a thermometer was used.

The sensor data in the PAMAP2 dataset was sampled at fixed-width sliding intervals of 1.28 seconds without any

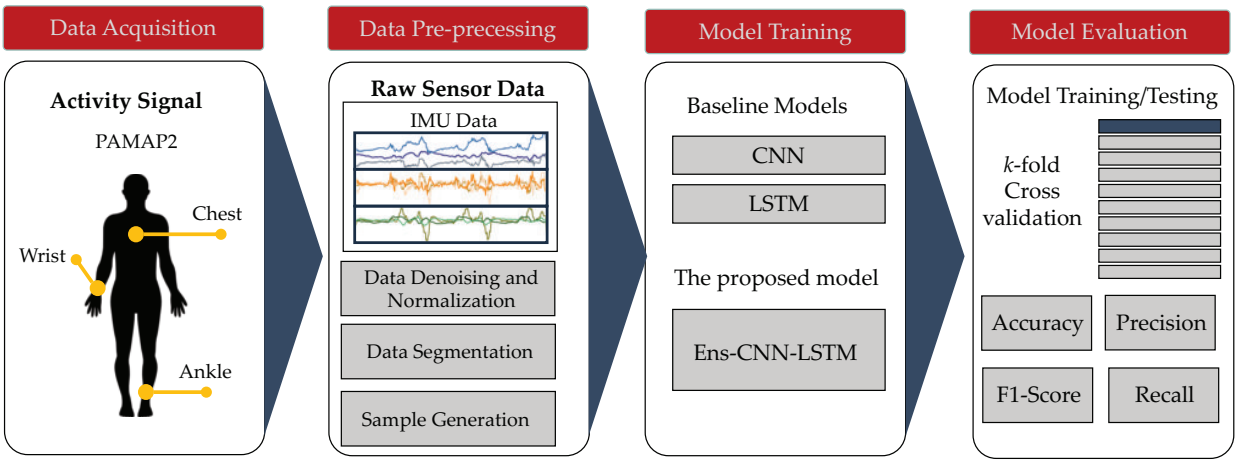


Fig. 1: S-HAR workflow used in this work.

overlap. Using this method, we obtained 128 measurements for each time interval, with the data being recorded at 100 Hz.

B. Data Augmentation

Data augmentation is a method applied to tackle the matter of class imbalance and scarcity of data in classification difficulties encountered in real-world scenarios. When there is a substantial disparity in the data quantity across classes, the classification method can show bias towards the class with more data, leading to reduced accuracy in classification. Data augmentation methods support balance the data distribution across different classes and alleviate this situation. Furthermore, when the dataset is limited in size, the data augmentation technique may be used to create artificial data that closely resembles the actual data. This can effectively enhance the effectiveness of the classification model. The research used the synthetic minority oversampling technique (SMOTE) [36] to create synthetic data by combining sample vectors between neighboring actual data vectors.

SMOTE is an oversampling strategy that uses distance-based techniques to increase the training data size. The method chooses an example of the minority class, labeled as x, from the training set and determines its K closest neighbors by calculating the shortest possible Euclidean distances between x and other examples of the same class. In order to generate a synthetic sample, denoted as x_{new}, the SMOTE technique randomly chooses one of the K closest neighbors from the minority class, which is labeled as x_k for the kth nearest neighbor. Subsequently, it calculates the disparity between x_k and x. The new synthetic sample, x_{new}, is obtained by multiplying the abovementioned variance by a randomly selected value ranging from 0 to 1, as shown in Equation (1). The synthetic instance, x_{new}, will be positioned on the line segment that connects x and x_k.

$$x_{new} = x + ||x - x_k|| \times rand(0, 1) \quad (1)$$

C. Ens-CNN-LSTM Architecture

The Ens-CNN-LSTM architecture is designed to combine the advantageous features of CNN and LSTM models to improve sensor-based HAR. This technique tries to efficiently collect spatial and temporal characteristics from the IMU sensor data.

Fig. 2 depicts the architecture of the Ens-CNN-LSTM model. The model has three essential parts:

1) The CNN obtains spatial characteristics from the input sensor data. CNNs excel in detecting intricate patterns and hierarchical characteristics within the given data.
2) The LSTM network is used to capture temporal relationships in the sequence of features recovered by CNN. LSTM models are suitable for analyzing time-series data and can capture and understand long-term relationships between data points.
3) Random forest: This ensemble learning technique is the ultimate classifier, merging the predictions generated by the CNN and LSTM elements.

The steps of the Ens-CNN-LSTM model are detailed as follows:

1) Initially, the input sensor data undergoes CNN processing to extract spatial features. The CNN architecture is depicted in Fig. 3.
2) The extracted features are fed into the LSTM network to capture temporal relationships. Fig. 4 presents the LSTM architecture.
3) The CNN and LSTM components generate predictions, referred to as P1 and P2 in Fig. 2. These predictions, along with a new feature set created by combining the CNN and LSTM outputs, are then input into a random forest classifier.
4) The random forest classifier takes the lead in this final step, producing the final prediction by integrating all the inputs it receives.

This ensemble technique aims to merge the advantages of deep learning architectures, namely CNN and LSTM, with

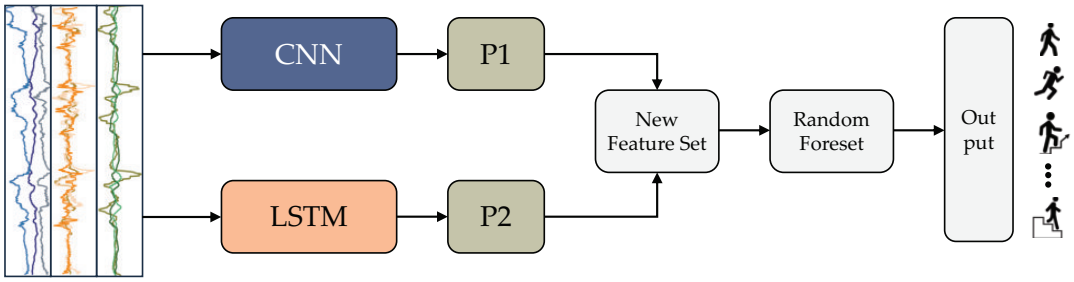


Fig. 2: Structure of the Ens-CNN-LSTM (P1 and P2 are the predictions from CNN and LSTM, respectively).

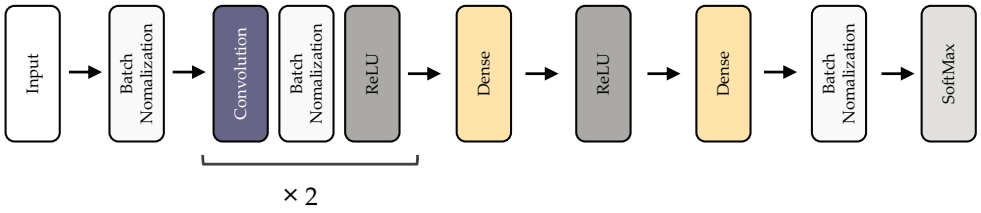


Fig. 3: Structure of the CNN.

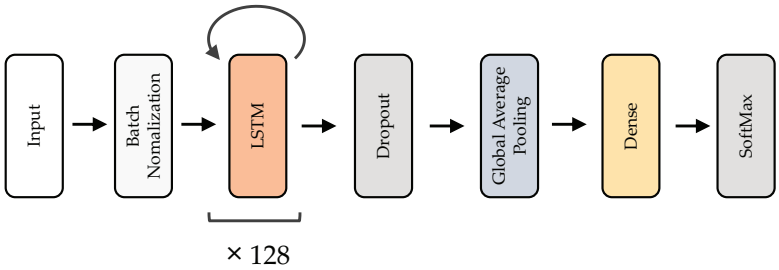


Fig. 4: Structure of the LSTM.

the resilience of the random forest algorithm. In comparison to standalone models, the implementation of the Ens-CNN-LSTM model is anticipated to provide superior accuracy and enhanced generalization in the identification of human actions from IMU sensor data.

IV. Experiments and Findings

The investigations in this research were performed using the Google Colab Pro platform, using a Tesla V100 GPU. The models were developed using Python 3.6.9, using the following essential libraries: TensorFlow 2.2.0, Keras 2.3.1, Scikit-Learn, Numpy 1.18.5, and Pandas 1.0.5.

We utilized the PAMAP2 dataset to train and assess our deep-learning models. The dataset was selected for its extensive scope, including recordings from nine persons involved in eighteen varied lifestyle activities. Data was gathered by using three Inertial Measurement Units (IMUs) placed on the dominant wrist, thorax, and ankle. Each IMU operated at a sampling rate of 100 Hz.

To ensure a robust and dependable review, we employed the 5-fold cross-validation technique. This method mitigates overfitting and provides a reliable assessment of the models' interpretation of unrecognized data.

Table I presents the performance metrics for the individual deep learning models (CNN and LSTM) and our proposed ensemble model (Ens-CNN-LSTM) on the PAMAP2 dataset.

TABLE I: Comparative performance metrics of individual and ensemble deep learning models on the PAMAP2 dataset using 5-fold cross-validation

Model	Recognition Effectiveness			
	Accuracy	Precision	Recall	F1-score
CNN	96.19%	96.19%	95.63%	95.84%
LSTM	97.35%	97.11%	97.06%	97.07%
Ens-CNN-LSTM	99.63%	99.61%	99.63%	99.61%

The findings indicate that our suggested Ens-CNN-LSTM model achieves outstanding efficiency compared to the individual CNN and LSTM models in all performance criteria. To be more precise:

- The CNN model had a accuracy rate of 96.19% and an F1-score of 95.84%.
- The LSTM model demonstrated improved effectiveness, with an accuracy rate of 97.35% and an F1-score of 97.07%.

- The Ens-CNN-LSTM model we suggested achieved outstanding outcomes, with an accuracy of 99.63% and an F1-score of 99.61%.

The findings emphasize the efficacy of our ensemble method in integrating the advantages of both CNN and LSTM architectures for HAR challenges.

Fig. 5 displays the confusion matrix for the Ens-CNN-LSTM model, offering a more comprehensive assessment of the model's effectiveness. This matrix provides a comprehensive analysis of the model's classification accuracy for various action classes, facilitating the detection of any possible misclassifications or difficult action kinds.

The outstanding effectiveness of the Ens-CNN-LSTM model could be credited to its capacity to efficiently collect spatial and temporal characteristics from the IMU sensor data. The CNN part is proficient in extracting spatial data, but the LSTM part efficiently records temporal connections. The random forest classifier integrates these complementing variables to generate the ultimate forecast, leading to enhanced accuracy and resilience compared to separate models.

V. Conclusion and Future Works

This study presents a new and innovative ensemble deep learning model, called Ens-CNN-LSTM, for recognizing human activities based on sensor data collected from IMU devices. Our methodology uses CNN's capabilities to extract spatial features and LSTM networks to capture temporal correlations. These components are then merged with a Random Forest classifier for reliable forecasts.

Our Ens-CNN-LSTM model, as demonstrated by experimental findings on the PAMAP2 benchmark dataset, surpasses the performance of individual CNN and LSTM structures. With an impressive accuracy of 99.63% and an F1-score of 99.61%, our model outperforms solo CNN and LSTM models by a significant margin. These results underscore the effectiveness of our ensemble method in extracting spatial and temporal characteristics from IMU sensor data, leading to improved accuracy in recognizing various physical activities.

Subsequent efforts will be directed toward improving the effectiveness and flexibility of the Ens-CNN-LSTM model in sensor-based HAR. This involves incorporating extra sensor modalities to obtain more details, investigating progressed data augmentation techniques to tackle class imbalance, creating personalization methods using transfer learning, maximizing the model for real-time processing on edge devices, broadening the scope of recognized actions, improving the comprehension of the model, and enhancing its capacity to handle noise and variations between individuals. The purpose of these developments is to develop a HAR system that is more inclusive, precise, and adaptable. This system may be used in many fields, such as healthcare tracking, athletics evaluation, intelligent settings, and human-computer interaction. By tackling these obstacles, we aim to expand the limits of sensor-based activity detection, eventually resulting in more significant applications in many areas related to human daily life and technological interactions.

Acknowledgment

This research project was supported by University of Phayao (Grant no. FF67-UoE-214); Thailand Science Research and Innovation Fund (Fundamental Fund 2024); National Science, Research and Innovation Fund (NSRF); and King Mongkut's University of Technology North Bangkok with Contract no. KMUTNB-FF-67-B-10.

References

[1] T. Magherini, A. Fantechi, C. D. Nugent, and E. Vicario, "Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living," *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 6, pp. 509–521, 2013.

[2] S. Mekruksavanich and A. Jitpattanakul, "Fallnext: A deep residual model based on multi-branch aggregation for sensor-based fall detection," *ECTI Transactions on Computer and Information Technology*, vol. 16, no. 4, pp. 352–364, 2022.

[3] S. Kalita, A. Karmakar, and S. M. Hazarika, "Human fall detection during activities of daily living using extended core9," in *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, 2019, pp. 1–6.

[4] Y.-L. Hsu, S.-C. Yang, H.-C. Chang, and H.-C. Lai, "Human daily and sport activity recognition using a wearable inertial sensor network," *IEEE Access*, vol. 6, pp. 31 715–31 728, 2018.

[5] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Recognizing and understanding sport activities based on wearable sensor signals using deep residual network," in *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2023, pp. 166–169.

[6] W. Du, "The computer vision simulation of athlete's wrong actions recognition model based on artificial intelligence," *IEEE Access*, vol. 12, pp. 6560–6568, 2024.

[7] S. Mekruksavanich, P. Jantawong, N. Hnoohom, and A. Jitpattanakul, "Recognizing driver activities using deep learning approaches based on smartphone sensors," in *Multi-disciplinary Trends in Artificial Intelligence*. Cham: Springer International Publishing, 2022, pp. 146–155.

[8] M. G. Morshed, T. Sultana, A. Alam, and Y.-K. Lee, "Human action recognition: A taxonomy-based survey, updates, and opportunities," *Sensors*, vol. 23, no. 4, 2023.

[9] S. Mekruksavanich and A. Jitpattanakul, "A residual deep learning method for accurate and efficient recognition of gym exercise activities using electromyography and imu sensors," *Applied System Innovation*, vol. 7, no. 4, 2024.

[10] P. N. Müller, A. J. Müller, P. Achenbach, and S. Göbel, "Imu-based fitness activity recognition using cnns for time series classification," *Sensors*, vol. 24, no. 3, 2024.

[11] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Deep learning approaches for har of daily living activities using imu sensors in smart glasses," in *2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2023, pp. 474–478.

[12] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[13] S. Mekruksavanich, D. Tancharoen, and A. Jitpattanakul, "A hybrid deep neural network with attention mechanism for human activity recognition based on smartphone sensors," in *2023 7th International Conference on Information Technology (InCIT)*, 2023, pp. 153–157.

[14] N. A. Choudhury and B. Soni, "Enhanced complex human activity recognition system: A proficient deep learning framework exploiting physiological sensors and feature learning," *IEEE Sensors Letters*, vol. 7, no. 11, pp. 1–4, 2023.

[15] X. Zhou, W. Liang, K. I.-K. Wang, H. Wang, L. T. Yang, and Q. Jin, "Deep-learning-enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6429–6438, 2020.

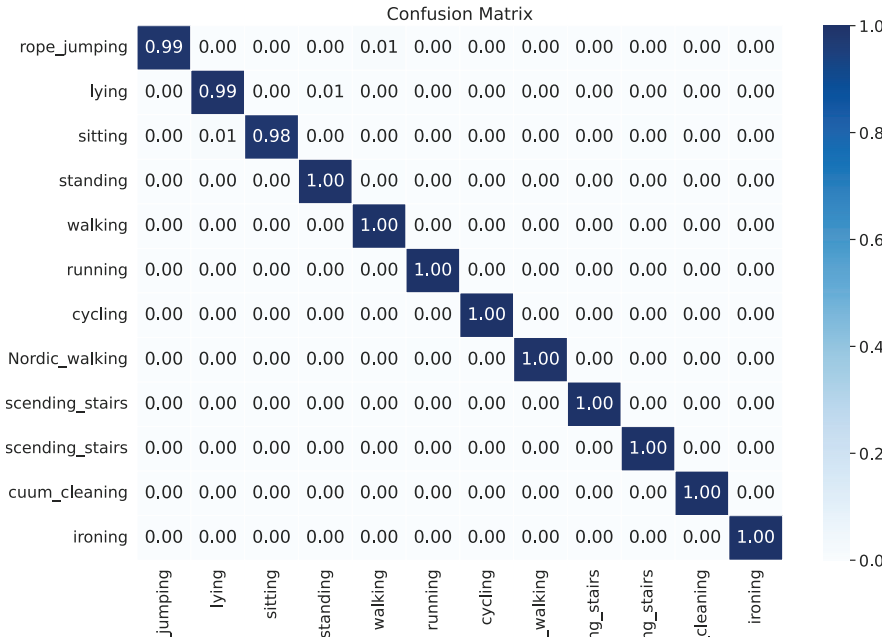


Fig. 5: Confusion matrix of the Ens-CNN-LSTM model for activity recognition on the PAMAP2 dataset, illustrating the classification performance across different physical activities.

[16] S. Mekruksavanich, P. Jantawong, and A. Jitpattanakul, "Attention based hybrid deep learning network for locomotive mode recognition in natural environments using wearable sensors," in *2023 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C)*, 2023, pp. 241–244.

[17] A. Rezaei, M. C. Stevens, A. Argha, A. Mascheroni, A. Puiatti, and N. H. Lovell, "An unobtrusive human activity recognition system using low resolution thermal sensors, machine and deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 70, no. 1, pp. 115–124, 2023.

[18] S. Mekruksavanich and A. Jitpattanakul, "Efficient recognition of complex human activities based on smartwatch sensors using deep pyramidal residual network," in *2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2023, pp. 229–233.

[19] S. Ankalaki, "Simple to complex, single to concurrent sensor-based human activity recognition: Perception and open challenges," *IEEE Access*, vol. 12, pp. 93 450–93 486, 2024.

[20] S. Pravesjit, P. Jantawong, A. Jitpattanakul, and S. Mekruksavanich, "Physique- based human activity recognition using deep learning approaches and smartphone sensors," in *2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, 2023, pp. 479–482.

[21] N. A. Choudhury and B. Soni, "An efficient and lightweight deep learning model for human activity recognition on raw sensor data in uncontrolled environment," *IEEE Sensors Journal*, vol. 23, no. 20, pp. 25 579–25 586, 2023.

[22] Z. A. Sunkad and Soujanya, "Feature selection and hyperparameter optimization of svm for human activity recognition," in *2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 2016, pp. 104–109.

[23] N. Tüfek and O. Özkaya, "A comparative research on human activity recognition using deep learning," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.

[24] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019, deep Learning for Pattern Recognition.

[25] S. Mekruksavanich, A. Jitpattanakul, P. Youplao, and P. Yupapin, "Enhanced hand-oriented activity recognition based on smartwatch sensor data using lstms," *Symmetry*, vol. 12, no. 9, 2020.

[26] Y. Kim and T. Moon, "Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 1, pp. 8–12, 2016.

[27] S. Mekruksavanich and A. Jitpattanakul, "Lstm networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, vol. 21, no. 5, 2021.

[28] Y. Mao, L. Yan, H. Guo, Y. Hong, X. Huang, and Y. Yuan, "A hybrid human activity recognition method using an mlp neural network and euler angle extraction based on imu sensors," *Applied Sciences*, vol. 13, no. 18, 2023.

[29] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, 2016.

[30] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[31] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, p. 1533–1540.

[32] H. Ullah and A. Munir, "Human activity recognition using cascaded dual attention cnn and bi-directional gru framework," *Journal of Imaging*, vol. 9, no. 7, 2023.

[33] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 351–360.

[34] W. Qi, H. Su, and A. Aliverti, "A smartphone-based adaptive recognition and real-time monitoring system for human activities," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 5, pp. 414–423, 2020.

[35] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 108–109.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, p. 321–357, jun 2002.

turnitin Page 192 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Design and Development of a Vertical Garden Station with Plants and an Automatic Fogging System for PM2.5 Reduction

Kanteera Mekruksavanich*, Natthayada Thamchaikul†, Parachaya Muentabutra‡
and Nongnapas Sutthipornmaneewat§
Bunyawat Witthayalai School
Lampang, Thailand
Email: *50069@bunyawat.ac.th; †50073@bunyawat.ac.th; ‡parachaya.19318@gmail.com
and §kim.nongnapas@gmail.com

***Abstract*—This paper aims to improve the efficiency of filtering airborne dust particles smaller than 2.5 microns (PM2.5) in a practical and applicable manner. The study explores three methods: a water spray system, a vertical garden system, and a combination. These systems, in a real-world scenario, transfer dust particles from a contaminated chamber to a second chamber. After 15 minutes, the water spray system reduced PM2.5 concentrations to 500 μg/m³. The vertical garden alone achieved a PM2.5 concentration of 404 μg/m³, while the combination of the spray system and vertical garden resulted in a PM2.5 concentration of 429 μg/m³. These results suggest that the vertical garden system is the most effective at lowering PM2.5 levels, providing a practical solution for air quality improvement. Testing on Golden Pothos leaves within the vertical garden revealed that unwashed leaves reduced PM2.5 levels to 201 μg/m³ within 75 minutes and to 223 μg/m³ after 93 minutes. However, when leaves were washed before filtering, PM2.5 levels decreased to approximately 74–115 μg/m³ within 73 minutes. This indicates that washing the leaves mainly improves the reduction of PM2.5 compared to using the vertical garden system without washing the leaves. The system also includes a PM2.5 detection component using an ESP8266 microchip connected to a sensor. This setup displays PM2.5 levels on an LED screen and sends alerts via the Blynk IoT application if levels exceed a preset threshold. The Arduino program can trigger alerts if PM2.5 levels surpass the defined limit. Tests on the mist spraying system showed it operates with 100% efficiency. In terms of cost-effectiveness, the vertical garden system is a more economical option than traditional air purifiers. It has a lower initial investment and a more affordable annual maintenance cost. Unlike conventional air purifiers, which incur higher yearly costs and require continuous operation to be effective, the vertical garden system provides ongoing air filtration even when not in use. This makes it a more budget-friendly choice for improving air quality, offering better long-term financial value.**

***Keywords*—PM2.5, airborne dust particle, vertical garden, Golden Pothos**

I. Introduction

Thailand is now experiencing an issue with dust particles smaller than 2.5 microns (PM2.5), which can be found in quantities that are above the stated health guidelines [1], [2]. The amount of dust significantly impacts the well-being of individuals, especially in Bangkok, a densely populated city that consistently experiences elevated levels of PM2.5 particles [3]. Exposure to PM2.5 dust can harm health due to the ability of these tiny particles to enter the respiratory tract and lungs, leading to significant health problems [4]. This is especially true for individuals who are more susceptible to harm, such as children, pregnant women, older adults, and those with pre-existing lung or cardiovascular diseases [5].

Air pollution, particularly the presence of PM2.5 , is one of the top environmental issues in the nation. These particles negatively affect human health [6], particularly impacting the lower respiratory system, lungs, and pulmonary alveoli, which are responsible for the exchange of oxygen and carbon dioxide in the bloodstream. Prolonged exposure to PM2.5 can lead to lung cancer. Additionally, it significantly impacts the lifestyle and living conditions of those exposed to it [7].

PM2.5 poses environmental challenges not only outdoors but also indoors, as contaminated air infiltrates buildings through doors and windows. Ji and Zhao [8] discovered a direct correlation between outdoor and indoor PM2.5 concentrations. If the PM2.5 levels are high outside, they are likely to be high inside as well. This situation affects building occupants, putting them at inevitable risk of PM2.5 exposure. Even with an air purifier operating for the standard 8-hour workday, PM2.5 levels remain elevated. It is crucial to monitor indoor air quality continuously and assess the potential health risks of inhalation during a building's use.

The majority of government agencies have implemented PM2.5 monitoring equipment in outdoor areas. However, systems are deficient for monitoring indoor air quality in residential buildings. These buildings, incredibly open structures such as education facilities, food courts, and residences, may have elevated levels of PM2.5, which poses a greater risk than enclosed constructions. Both governmental organizations and academic institutions have attempted to find solutions, but these efforts have not been successful. Even when the government declared it a national priority, the issue remained unresolved. Therefore, finding long-term strategies and tactics

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

to address this problem is crucial [9].

Various methods exist today to mitigate PM2.5, including plants and air purifiers. However, air purifiers are costly and have a limited lifespan, while plants require extensive maintenance. Consequently, this work proposed developing a vertical garden system that can efficiently and automatically eliminate dust particles smaller than 2.5 microns.

This study aims to achieve the following contributions:

- To investigate an optimal design that enhances the efficiency of capturing dust particles smaller than 2.5 microns.
- To create a vertical garden system specifically designed to reduce PM2.5 levels.

II. Related Works

The relationship between exterior environmental elements and interior air quality has been a central focus in indoor air quality (IAQ) research. Multiple studies have conclusively linked indoor air pollution to a wide array of health problems, including respiratory ailments, cardiovascular troubles, skin conditions, neurological disorders, and many forms of cancer [10]. Contrary to the notion that indoor air is naturally cleaner than outside air, substantial studies have shown that pollution levels inside maybe 2-5 times greater than outdoors [11]. Particles of a diameter of less than 2.5 microns have been recognized as significant contributors to adverse health impacts, particularly affecting the respiratory system.

The proliferation of Internet of Things (IoT) devices has significantly improved the domain of air quality tracking [12]. IoT for identifying and assessing air quality has been a central focus of recent breakthroughs [13]. The combination of IoT and cloud computing transforms the monitoring method of indoor air quality by offering more dynamic and real-time data [14]. Of particular significance in this context is the use of air quality sensors. These sensors, adept at detecting a variety of chemicals in the environment, are integrated with microcontrollers for the purpose of data analysis and monitoring [15]. This technical setup forms the backbone of our advanced air quality tracking system.

The actual implementation of IoT in evaluating air quality involves using Arduino-based systems for tracking several air characteristics, including gaseous amounts, via a serial surveillance method. A notable research study conducted by Gupta et al. [16] demonstrated the enormous promise of the IoT in tracking pollution in the environment. The research included the development of an appliance that constantly monitors several environmental factors such as humidity, temperature, PM2.5, and PM10. This system thoroughly explains the prevailing environmental circumstances. In order to expand the range of applications for the IoT for measuring IAQ, Saini et al. [17] created a system that uses a wireless sensor network. This system allows for continuous surveillance of air quality in different places and provides actual time accessibility to the data through tablets, smartphones, and other internet-enabled gadgets.

These advancements highlight IoT's significant influence in IAQ tracking, providing creative approaches and improving the capacity to detect and address environmental problems efficiently.

III. Materials and Methods

This research proposes guidelines for developing an IoT indoor air quality system that tracks users' PM2.5 levels and features a plants vertical garden station with an automatic fogging system to reduce PM2.5 levels. The details of the materials and procedures are provided below.

A. Materials

The equipment used in this investigation consists of the following:

1) PVC pipe frame: 1.50 meters wide, 1.80 meters high, and 2.00 meters long
2) Plastic sheet: 3 meters wide and 6 meters long, quantity: 1 sheet
3) Exhaust fan: 13.5 cm, quantity: 1 unit
4) Golden Pothos plants: 40 plants
5) Water bottles: 1.5 liters, quantity: 30 bottles
6) Golden pothos-pot racks: 1 set
7) Misting system: 1 set
8) PM2.5 particulate matter meters: 2 units
9) ESP 8266 board: 1 unit
10) PM2.5 sensors: 1 piece
11) LED screen: 1 unit

B. Methods

1) Investigating Optimal Designs to Enhance the Efficiency of PM2.5 Dust Particle Collection:

1) Construct a laboratory to capture PM2.5 dust particles. The laboratory should measure 1.5 m × 2 m × 1.8 m and consist of two connected chambers, as illustrated in Fig. 1.
2) Construct a vertical garden measuring 0.9 m × 1.5 m using Golden Pothos, as depicted in Fig. 2. Set up a test chamber to evaluate the effectiveness of capturing PM2.5 particles.
3) Generate PM2.5 from incense to reach a target concentration of 500 μg/m3. Activate the exhaust fan to transfer PM2.5 to the second adjoining chamber.
4) Assess the efficiency of the water spray system in capturing PM2.5 particles for 15 minutes. Continuously record the PM2.5 levels at 5-minute intervals during this period.
5) Repeat step 3 twice, altering the dust capture method first to the vertical garden system and then to a combination of the water spray system and the vertical garden.

2) Comparing the Efficiency of PM2.5 Dust Particle Trapping in Vertical Plantations with and without Leaf Washing:

1) Prepare the test chamber to assess the efficiency of dust collection for PM2.5 particles. Generate PM2.5 from incense to reach a concentration of 500 μg/m3. Activate

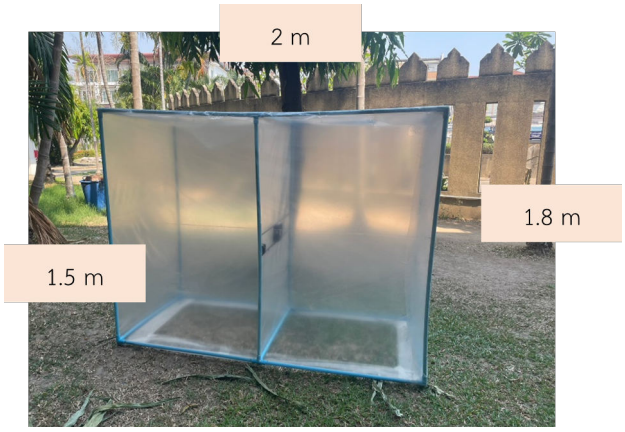


Fig. 1: A laboratory setup for capturing PM2.5 dust particles.

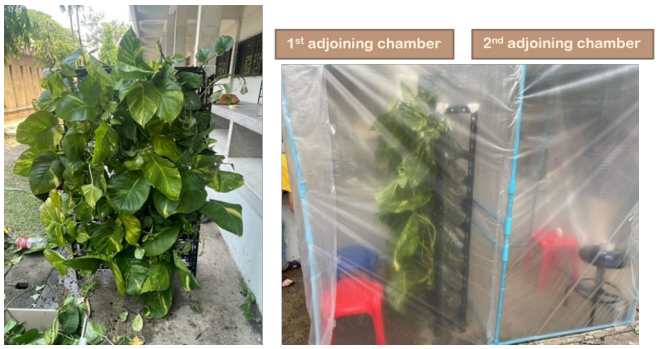


Fig. 2: Laboratory setup with vertical plantations.

the exhaust fan to move PM2.5 to the second chamber. During this experiment, the misting system will remain off to avoid washing the leaves.

2) Measure the treated particulate matter in the second chamber for 1 hour and 44 minutes.
3) Repeat steps 1 and 2 for three additional cycles, this time activating the leaf washing system every 30 minutes for 1 minute each time.

3) Building a Vertical Garden System for PM2.5 Particulate Matter Absorption: Develop a program for measuring PM2.5 particulate matter using the ESP 8266, a mini WiFi microchip. Connect this to a PM2.5 sensor and display the PM2.5 readings on an LED screen. Configure the system to send alerts via the Blynk IoT application when PM2.5 levels exceed a specified threshold. The Arduino programming platform will be used to implement this system. The procedure for notifying when PM2.5 levels exceed the specified limit is illustrated in Fig. 3.

Alerts are sent via the application in a vertical garden made up of Golden Pothos (used to capture PM2.5 particles), a water spray system (for washing the leaves), and a PM2.5 measurement system (which can set the PM2.5 meter to start and stop notifications of the PM2.5 level). Additionally, the device measuring the PM2.5 levels emits a louder sound, as shown in Fig. 3.

4) Performance Evaluation of the Vertical Garden System for PM2.5 Particulate Matter Absorption:

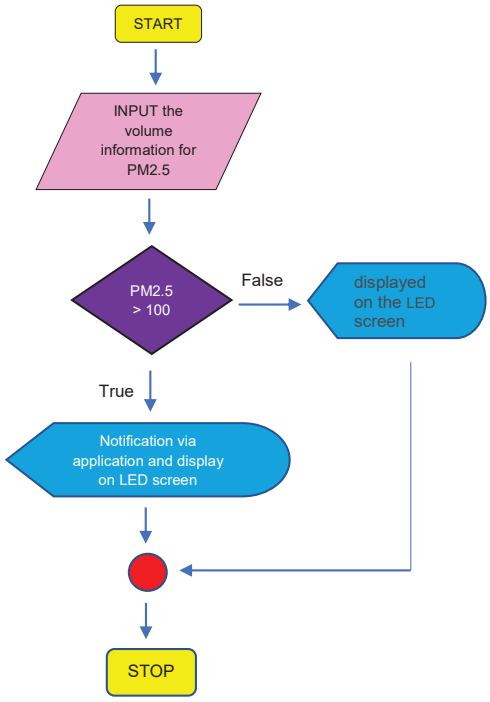


Fig. 3: The procedure for notifying when PM2.5 levels exceed the specified limit.

1) Test the functionality of the sensor to measure PM2.5 dust particles:
   a) Set the machine to trigger an alert when PM2.5 levels reach 300 µg/m³ and to alert again when the levels drop to 200 µg/m³.
   b) Introduce incense smoke to the machine and observe the alerts, both from the device's sound notifications and the application. Repeat this experiment 10 times.
2) Conduct a comparative study on the cost-effectiveness of vertical garden systems in absorbing PM2.5 dust particles versus various brands of air purifiers available on the market.

IV. Results and Discussion

A. Investigating Optimal Models to Enhance the Efficiency of PM2.5 Particle Collection

The PM2.5 levels transferred from a dust-contaminated chamber to a second chamber using various methods—such as the water spray system, vertical garden system, and a combination of both—can be illustrated in a comparative chart for each method, as shown in Fig. 4.

The results demonstrate the PM2.5 levels transferred from the dust-contaminated chamber to the second chamber using three different methods: a water spray system, a vertical garden system, and a combination of both. After 15 minutes, the water spray system alone resulted in a PM2.5 concentration of 500 µg/m³, while the vertical garden system alone achieved a PM2.5 concentration of 429 µg/m³. The combined water spray

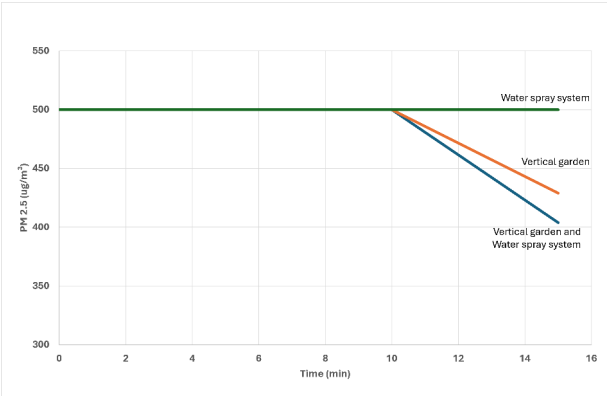


Fig. 4: The relation between the level of PM2.5 and time.

and vertical garden system resulted in a PM2.5 concentration of 404 µg/m³. These findings suggest that the vertical garden system is the most effective method for reducing PM2.5 levels.

B. Comparing the Dust Particle Trapping Efficiency of a Vertical Garden With and Without Leaf Washing for PM2.5

The results show the PM2.5 levels released from a dust-contaminated chamber to a second chamber using a vertical garden, comparing washed and unwashed leaves, as shown in Fig. 5 and Fig. 6, respectively. According to the graph, for the vertical garden with unwashed leaves, PM2.5 levels decreased most significantly at 75 minutes, dropping to 201 µg/m³ and further to 223 µg/m³ after 93 minutes.

In contrast, PM2.5 levels showed a more substantial reduction for the vertical garden with washed leaves. In the first experiment, the level decreased to 74 µg/m³ after 73 minutes, and across all three experiments, the concentration reduced to 115 µg/m³ after 60 minutes and remained stable at this level. After 93 minutes, the concentration further decreased to 95 µg/m³.

The comparison of PM2.5 levels in the vertical garden system, with and without leaf washing, demonstrates that the system with leaf washing achieved a more significant reduction in PM2.5 levels than the system without leaf washing.

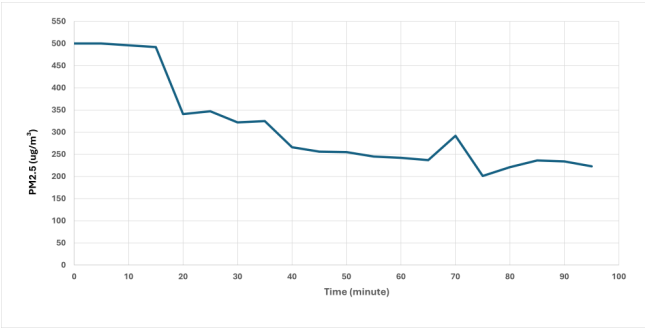


Fig. 5: PM2.5 levels in the second room using a vertical garden with unwashed leaves.

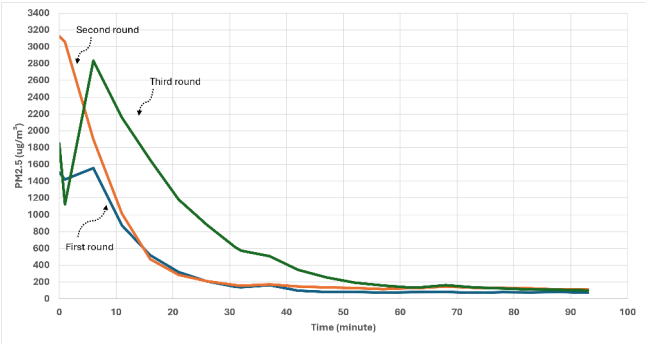


Fig. 6: PM2.5 levels in the second room using a vertical garden with washed leaves.

C. Evaluating the Efficiency of a Vertical Garden System in Capturing PM2.5 Particles and Comparing Its Cost-Effectiveness with Various Air Purifier Brands on the Market

1) Assessing the Efficiency of the Vertical Garden System in Capturing PM2.5 Particles: The study on dust trapping efficiency involves testing the sensor's ability to measure dust. This is done by setting the machine to alert when the PM2.5 dust level reaches 300 µg/m³. The machine will alert again when the PM2.5 level drops to 200 µg/m³. Incense smoke is then introduced to the machine, and notifications are monitored by listening to the sounds from the device and checking the alerts in the application.

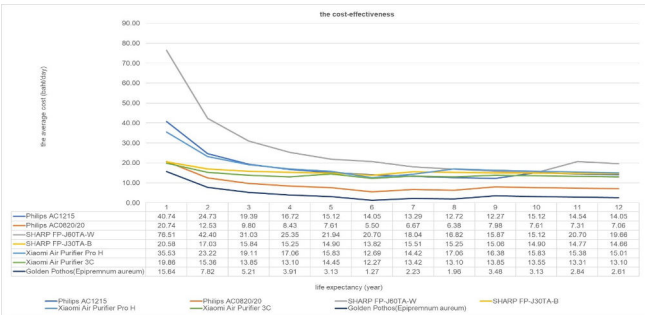


Fig. 7: The cost-effectiveness of vertical garden systems absorbing PM2.5 with various brands of air purifiers on the market.

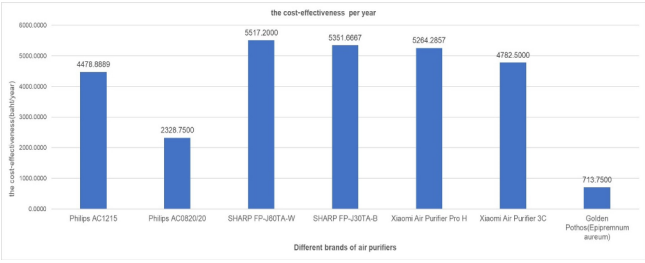


Fig. 8: The cost-effectiveness analysis of vertical garden systems with various brands of air purifiers on the market.

2) Comparing the Cost-Effectiveness of the Vertical Garden System for Capturing PM2.5 Dust with Various Market Air Purifiers: As the results show in Fig. 7 and Fig. 8, a study of various air purifier dealers' websites shows that the average cost of the six purifiers from three different brands is significantly higher than that of a vertical garden. Although the Xiaomi Air Purifier 3C is less expensive than a vertical garden, it has a shorter service life and requires twice the number of filters.

A vertical garden has an initial cost and an average annual cost lower per day than similar costs of other purifiers, such as the Philips AC0820/20. The difference in cost is notable when considering annual expenses. Moreover, a vertical garden can continuously filter air even when not actively powered, unlike other air purifiers that need to be turned on to filter air. This feature can further reduce the overall cost. Therefore, the vertical garden innovation is more cost-effective than the air purifiers studied.

V. Conclusion

The project focused on enhancing the efficiency of a vertical garden system for capturing dust particles smaller than 2.5 microns (PM2.5). Various methods were assessed, including a water spray, vertical garden, and combination. This evaluation involved directing smoke from incense through these systems from a dust-laden chamber to a secondary chamber. After 15 minutes, the water spray system achieved a PM2.5 concentration of 500 µg/m³. The vertical garden system reduced PM2.5 to 404 µg/m³, while the combination of water spray and vertical garden resulted in a PM2.5 level of 429 µg/m³. These results indicate that the vertical garden system is the most effective at reducing PM2.5 levels, as the water spray system alone is less capable of capturing such small particles. The water spray method is better suited for larger dust particles.

Additionally, tests compared the efficiency of the vertical garden system with washed versus unwashed Golden Pothos leaves. With unwashed leaves, PM2.5 levels decreased to 201 µg/m³ within 75 minutes and 223 µg/m³ after 93 minutes. In contrast, using washed leaves led to a more significant reduction, with PM2.5 levels dropping to 74–115 µg/m³ within approximately 73 minutes. This demonstrates that washing the leaves improves filtration efficiency by removing accumulated dust. With its textured leaves, the Golden Pothos was highly effective in reducing dust, outperforming other plant species tested.

Furthermore, a monitoring system utilizing an ESP8266 microchip and a PM2.5 sensor was developed. Using the Arduino program, this system provides real-time PM2.5 level readings and sends alerts when levels exceed a set threshold. The system proved highly effective in both monitoring and notification functions.

In terms of cost-effectiveness, the vertical garden system emerged as a more economical choice compared to traditional air purifiers. Despite the air purifier's slightly lower annual operational cost, it requires continuous use to be effective. In contrast, the vertical garden system can filter air even when not in operation, offering better overall value. This reassures the audience that the vertical garden system is a more cost-effective solution in the long run, particularly when PM2.5 levels are not extremely high.

References

[1] I. Bensalam, R. Saelim, A. Samoh, N. Kongbok, I. Toheng, and S. Musikasuwan, "An investigation on the influence of meteorological factors on pm2.5 concentration: Towards predictive models for songkhla, thailand," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 2024, pp. 1–5.

[2] T. Thanavanich, M. Yaibuates, and P. Suchaya, "Improving the accuracy of forecasting pm2.5 concentrations with hybrid neural network model," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 2021, pp. 18–22.

[3] A. Kodaka, N. Leelawat, J. Tang, Y. Onda, and N. Kohtake, "Status of industrial complex activity explained by air quality: Central thailand," in *2023 Third International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, 2023, pp. 123–126.

[4] W. Wongchan, "Health promotion and prevention diseases of particulate matter (pm2.5) in school-age children," *Ramathibodi Medical Journal*, vol. 46, no. 4, pp. 52–65, Dec. 2023.

[5] M. Oprea and H.-Y. Liu, "A knowledge based approach for pm2.5 air pollution effects analysis," in *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2016, pp. 1–8.

[6] Y. Li, Z. Chen, and J. Li, "How many people died due to pm2.5 and where the mortality risks increased? a case study in beijing," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 485–488.

[7] L.-J. Chen, Y.-H. Ho, H.-C. Lee, H.-C. Wu, H.-M. Liu, H.-H. Hsieh, Y.-T. Huang, and S.-C. C. Lung, "An open framework for participatory pm2.5 monitoring in smart cities," *IEEE Access*, vol. 5, pp. 14 441–14 454, 2017.

[8] W. Ji and B. Zhao, "Contribution of outdoor-originating particles, indoor-emitted particles and indoor secondary organic aerosol (soa) to residential indoor pm2.5 concentration: A model-based estimation," *Building and Environment*, vol. 90, pp. 196–205, 2015.

[9] J. J. R. Balbin, A. J. G. De Guzman, and C. J. C. Rambuyon, "Air pollutant detection system utilizing an iot-based electronic nose for air purifier," in *2022 6th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*, 2022, pp. 136–140.

[10] S. Zhong, Z. Yu, and W. Zhu, "Study of the effects of air pollutants on human health based on baidu indices of disease symptoms and air quality monitoring data in beijing, china," *International Journal of Environmental Research and Public Health*, vol. 16, no. 6, 2019.

[11] G. Settimo, M. Manigrasso, and P. Avino, "Indoor air quality: A focus on the european legislation and state-of-the-art research in italy," *Atmosphere*, vol. 11, no. 4, 2020.

[12] J. H. Buelvas P., F. E. Avila B., N. Gaviria G., and D. A. Munera R., "Data quality estimation in a smart city's air quality monitoring iot application," in *2021 2nd Sustainable Cities Latin America Conference (SCLA)*, 2021, pp. 1–6.

[13] G. Parmar, S. Lakhani, and M. K. Chattopadhyay, "An iot based low cost air pollution monitoring system," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, 2017, pp. 524–528.

[14] J. Jo, B. Jo, J. Kim, S. Kim, and W. Han, "Development of an iot-based indoor air quality monitoring platform," *Journal of Sensors*, vol. 2020, no. 1, p. 8749764, 2020.

[15] S. Malleswari and T. K. Mohana, "Air pollution monitoring system using iot devices: Review," *Materials Today: Proceedings*, vol. 51, pp. 1147–1150, 2022, cMAE'21.

[16] H. Gupta, D. Bhardwaj, H. Agrawal, V. A. Tikkiwal, and A. Kumar, "An iot based air pollution monitoring system for smart cities," in *2019 IEEE International Conference on Sustainable Energy Technologies and Systems (ICSETS)*, 2019, pp. 173–177.

[17] J. Saini, M. Dutta, and G. Marques, "A comprehensive review on indoor air quality monitoring systems for enhanced public health," *Sustainable Environment Research*, vol. 30, no. 1, p. 6, Jan 2020.

turnitin Page 197 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

Reel Tower Control using Machine Learning

MyeongSu Jeong
SW Research
GyeongBuk Research Institute of Vehicle Embedded Technology
Yeongcheon-si, Republic of Korea
ms.jungn@givet.re.kr

ChangSoo Moon
SW Research
GyeongBuk Research Institute of Vehicle Embedded Technology
Yeongcheon-si, Republic of Korea
cs.moon@givet.re.kr

JaeHoon Chung
Reserch Institute
YJ Link
Daegu
ohlyes@naver.com

***Abstract*— In this paper, we propose a machine learning-based approach to streamline the loading and unloading processes of electronic components in reel towers. The conventional method typically involves loading electronic components into nearby empty spaces without considering their frequency of use, resulting in increased robot movements and longer processing times. Our method utilizes machine learning to predict the interval between component requests and strategically load components based on their anticipated usage frequency. Experimental results demonstrate that our approach significantly reduces the number of robot movements compared to the conventional method. Future work will involve implementing our model in real reel tower settings to further validate its effectiveness and optimize its performance. Through these efforts, we aim to introduce more efficient management practices for electronic component storage and retrieval in reel towers, ultimately enhancing overall productivity and reducing operational costs.**

Keywords—SMT, SMD, PCB, Machine learning, Reel tower

I. INTRODUCTION

Surface mount technology (SMT) is a pivotal technique in electronics manufacturing, integral for affixing electronic components onto printed circuit boards (PCBs). Surface mount devices (SMDs), including resistors, capacitors, and inductors, are directly bonded to PCB surfaces. These SMDs are packaged in reels and loaded in specialized load units known as reel towers. Reel towers serve as essential repositories for PCB manufacturing and loading purposes. Employing an SMD reel tower streamlines reel identification and enables dynamic management. The reel tower management process entails two primary tasks: loading new electronic component reels into the tower and unloading requested electronic components. This process is facilitated by a pickup robot stationed at the reel tower entrance, navigating between load and unload areas as directed by operators. However, significant movement time is incurred if the load and unload areas are distant from the pickup robot.

Recently, there has been a rise in the proposal of smart process technologies integrating automation to mitigate the costs associated with managing and maintaining large quantities of electronic components[1][2]. In this paper, we introduce a method for automating the loading and unloading of electronic components from reel towers using robots, aiming to minimize the number of robot movements and enhance efficiency. Specifically, we leverage predictions from a machine learning model to optimize the placement of required electronic components within the reel tower. Machine learning models excel at discerning patterns from extensive datasets, enabling them to forecast future trends in electronic component load and unload. Our trained model anticipates the next unload time for electronic components requested for loading. Based on these predictions, electronic components are strategically loaded in the most optimal locations within the reel tower. The machine learning model employed utilizes recurrent neural networks (RNNs), long short-term memory (LSTM), and gated recurrent units (GRUs). We propose a methodology to streamline the number and duration of robot movements by comparing the performance of these algorithms in predicting component requests. Additionally, this paper seeks to address the challenges inherent in the implementation of automated reel towers in various industrial settings. A key aspect of this technology is its ability to adapt to diverse manufacturing environments, each with unique constraints and requirements. By evaluating the performance of RNN, LSTM, and GRU models, the proposed system aims to dynamically adjust to varying patterns of component usage and demand, thereby enhancing its robustness and reliability.

This paper is structured as follows: Chapter 2 provides background information, while Chapter 3 delves into the results obtained from the application of the machine learning model. Finally, Chapter 4 offers concluding remarks and outlines avenues for future research.

II. METHOD

A. Background

In this chapter delineates the foundational background of reel towers, serving as repositories for reels housing surface mount device (SMD) electronics. Two primary types of reel towers are identified: cylindrical and orthogonal. The cylindrical variant boasts a simplified structure, obviating the necessity for an X-axis transfer device, with the length of the electronic component pickup system affixed to the inner radius of the reel compartment. Figure 1 shows a cylindrical reel tower, while Figure 2 shows an orthogonal reel tower.

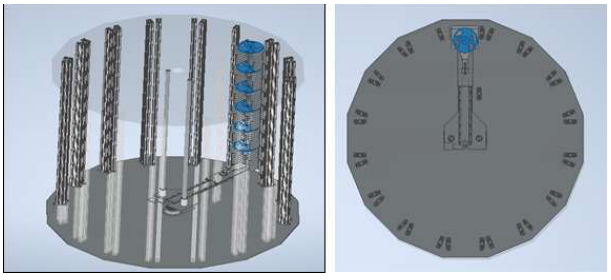


Fig. 1. Structure of cylindrocal reel tower.

979-8-3503-9174-9/24/$31.00 ©2024 IEEE

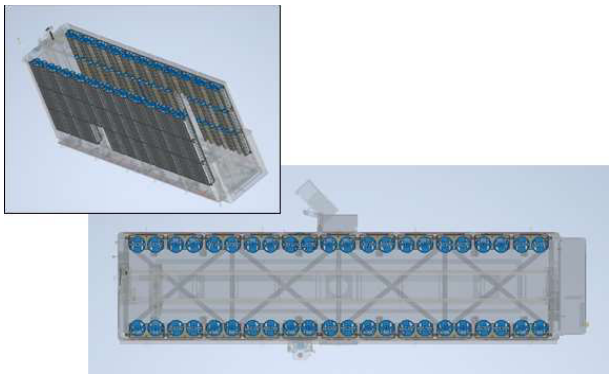


Fig. 2. Structure of orthogonal reel tower.

The orthogonal reel tower design offers a 30% larger stacking area for electronic components per reel within the same machine area as a cylindrical tower. Within the electronics compartment, the width of the cylindrical tower increases from the entrance towards the interior, while the width of the orthogonal tower remains constant from the entrance, minimizing wasted space. In contrast to the cylindrical configuration, which exhibits square proportionality when expanded, the rectangular shape maintains simple proportionality. The orthogonal type features electronic component compartments arranged in parallel, facing each other at regular intervals, with an electronic component pickup system installed between them, determining the width of the reel tower.

In orthogonal reel tower systems, pickup robots are required to cover minimal distances, which demands a short system length. Additionally, they need to possess rapid acceleration and deceleration capabilities to enable high-speed movements for productivity and short-distance reciprocation

Figure 3 shows the process of loading a newly issued reel unit of electronics into the reel tower. The Reel main control PC is responsible for managing and controlling the electronic components throughout this process. It acquires an ID from the Manufacturing Execution System (MES) server upon receiving an electronic component request. The MES server, overseeing the entire manufacturing process, generates an ID and transmits it to the Reel main control PC. Electronic components with newly assigned IDs are placed into temporary containers and subsequently transported to the reel tower. The relocated electronic components are positioned near the entrance and loaded within the reel tower by a pickup robot. Upon request from a PCB for the necessary electronic component, the pickup robot unloads the component's ID and transports it to the load area.



Fig. 3. Sequence of issued reel ID

B. Training data

Training data is generated by simulating load and unload scenarios involving electronic components utilized across five distinct PCBs. The cumulative count of components across these PCBs amounts to 366. Figure 4 shows the top 20 components, sorted in descending order based on their frequency of requests.

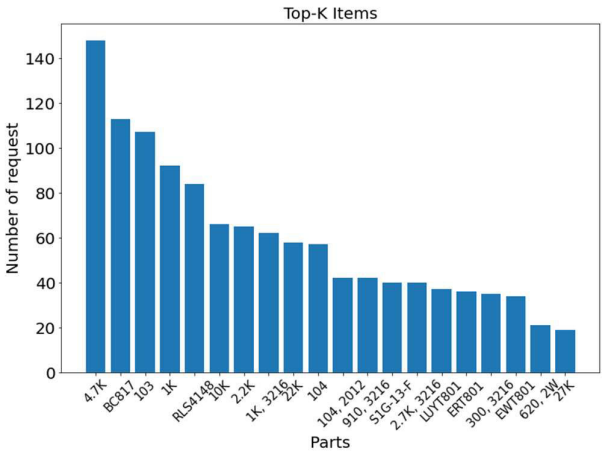


Fig. 4. Top 20 parts by number of part request

Requests for electronic components for PCB manufacture vary from one type to another, but frequently used electronic components are also common across different PCB manufactures and have a high likelihood of being loaded and unloaded. This paper leverages this phenomenon to predict the time required for loading and unloading electronic components. Employing a sliding window technique, electronic components are loaded over time in the order of request, with the model input size set to 32. Figure 5 shows the process of generating electronic component request data using a sliding window approach.

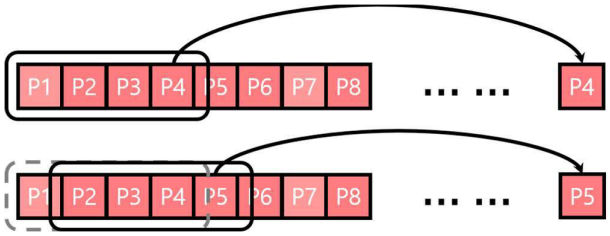


Fig. 5. Generation of parts request data using sliding window

C. Model selection

Sequential data is characterized by the arrangement of objects in a specific order within a dataset, where any alteration to this order results in a loss of its distinctive characteristics. Analyzing sequential data requires an initial step to represent its temporal properties and subsequently devise methods for predicting information from it. In the realm of machine learning algorithms, there has been a proposal to train extensive datasets while preserving the sequential order within artificial neural networks. Given the variability in the request interval of electronic components over time, our objective is to forecast and compare these intervals utilizing recurrent neural network (RNN), long short-term memory (LSTM), and gated recurrent unit (GRU) models. Figure 6 shows the RNN, Figure 7 shows the GRU, and Figure 8 shows the LSTM model structure.

turnitin Page 199 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

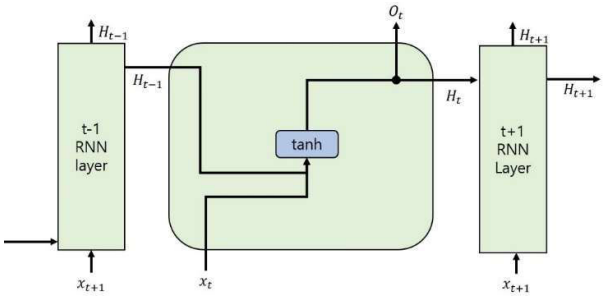


Fig. 6. RNN architecture

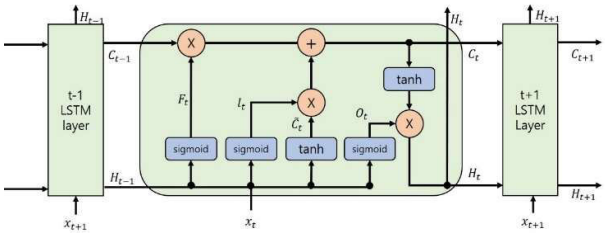


Fig. 7. LSTM architecture

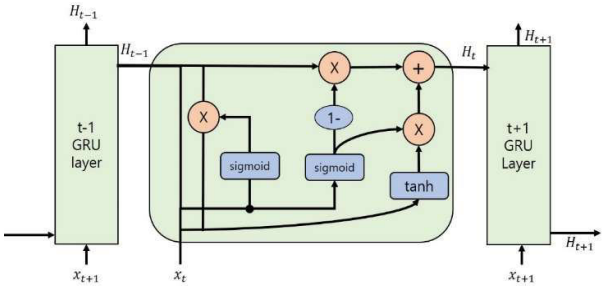


Fig. 8. GRU architecture

RNNs are models that incorporate previous states when computing a state for a given input [3]. They are particularly suited for sequence prediction tasks because they maintain a memory of previous inputs in the sequence. However, RNNs suffer from the vanishing gradient problem, which makes it difficult for them to capture long-term dependencies in sequences. This limitation often results in poor performance when modeling long-range dependencies, necessitating the development of more advanced architectures like LSTMs and GRUs to address these challenges. LSTMs are models capable of effectively connecting distant memories and current states by utilizing memory cells within the network, which employ the hidden layer as the subsequent input [4]. The architecture of LSTMs includes special units called memory cells that can maintain their state over long periods. These cells use gates (input, output, and forget gates) to control the flow of information, allowing them to learn which data in a sequence is important to keep or discard. This design helps LSTMs overcome the vanishing gradient problem and makes them effective for tasks requiring long-term memory, such as predicting the intervals between requests for electronic components in a reel tower. The flexibility and robustness of LSTMs make them a suitable choice for optimizing the placement of electronic components by accurately predicting future requests. GRU is a model designed to streamline computation by reducing the parameters of the model through the elimination of memory cells found in LSTMs [5]. GRUs simplify the LSTM architecture by combining the forget and input gates into a single update gate and merging the cell state and hidden state. This reduction in complexity leads to faster training times and fewer computational resources while maintaining similar performance. In the context of reel towers, GRUs provide an efficient alternative for predicting the load and unload times of electronic components, thereby optimizing the robot's movement patterns. The balance between simplicity and performance in GRUs makes them an attractive choice for applications where computational efficiency is critical.

In summary, RNNs, LSTMs, and GRUs each have unique characteristics that make them suitable for different aspects of sequence prediction tasks. RNNs are effective for short-term dependencies, LSTMs excel at capturing long-term dependencies, and GRUs offer a good trade-off between performance and computational efficiency. By comparing these models, we aim to determine the most effective algorithm for minimizing robot movements and enhancing the efficiency of the reel tower management process

In this paper, both the currently requested part and the previously requested part are fed into the embedding layer and then processed through the model cell. The final cell within the model predicts the release time of the currently requested electronic part. By comparing RNN, LSTM, and GRU models, we aim to identify the algorithm that most effectively reduces the number of robot movements required for loading and unloading electronic components in a reel tower, thereby enhancing overall operational efficiency.

D. Batch algorithm

Since the value predicted by the model is the next time the electronic component will be unloaded, it is necessary to utilize this prediction to decide where to load the electronic component in the reel tower. In this paper, we divide the reel tower space into classes based on the percentile of request times for loading electronic components on a reel-by-reel basis. Electronic components that will be unloaded in the near future are loaded in the class with the shortest unload movement radius. Conversely, electronic components that will be unloded in the far future are loaded in the class with the longest unload movement radius. This process reduces the movement distance of the pickup robot for loading and unloading electronic components and enables efficient deployment.

III. RESULT

In this chapter, we assess and interpret both existing methods and our proposed approach. The system specifications employed in our experiments are as follows:

TABLE I. SYSTEM SPECIFICATION

CPU	12th Gen Intel Core i5-13400F 2.5GHz
RAM	32GB
OS	Windows 10 Pro

A. Result of training

Figures 9 and 10 shows graphs depicting the calculated loss value per epoch. It is evident that the loss value decreases progressively throughout the training process. For a more detailed assessment of accuracy, please refer to Figure 11, which demonstrates the variance between the actual and predicted values.

turnitin Page 200 of 204 - Integrity Submission Submission ID trn:oid:::1:3308505772

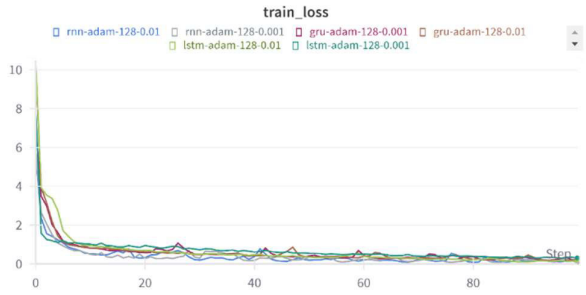


Fig. 9. Results of training loss during training epochs

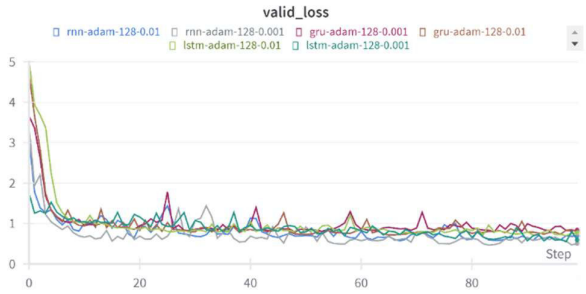


Fig. 10. Results of validation loss during training epochs

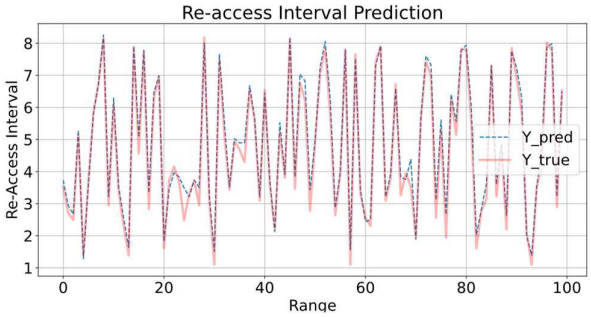


Fig. 11. Real values(Y_ture) and predicted values(Y_pred)

B. *Result of simulation*

Next, we employ the trained model to quantify the number of coordinate movements made by the robot. Each move signifies a shift of one unit along the X and Y coordinates in the two-dimensional space. There are 366 types of components utilized across 5 PCBs, with a total of 3,663 load and unload requests for each electronic component type. The reel tower dimensions are set at 40×10. The conventional method entails loading reels in proximity to the entrance when space is available (first in, first out), whereas the approach advocated in this paper involves loading reels based on the interval predicted by the model between future requests. The existing method follows a first-in-first-out (FIFO) approach, where the reel is loaded when an empty space exists close to the entrance. In contrast, the method proposed in this paper is to load the reel according to the interval after the model predicts the interval between future requests.

TABLE II. NUMBER OF MOVES AND PERCENT REDUCTION BY MODEL

Model	FIFO	RNN	LSTM	GRU
Number of moves	66,122	41,486	40,206	40,812
Rate of reduction(%)	-	37	39	38

IV. CONCLUSIONS

In this paper, we presented an approach to efficiently manage the load and unload of electronic components in a reel tower by employing machine learning techniques. The conventional method involves loading newly arrived electronic components in nearby empty spaces without considering their frequency of use. However, this approach is suboptimal as it results in increased robot movements and longer load and unload times.

Our proposed method utilizes machine learning to discern the usage frequency of electronic components and predict the interval between requests. This enables us to minimize the number of robot movements and reduce the time required for load and unload operations in loading and unloading electronic components.

In future work, we aim to implement the proposed model in an actual reel tower to validate its efficacy in reducing the number of pickup robot movements for load and unload operations based on component usage frequency compared to the conventional method. By deploying the three proposed models in a real-world reel tower setting, we intend to identify the model that minimizes pickup robot movements and further refine the algorithm accordingly.

Through the reduction of movements and time required by the pickup robot, we anticipate that the reel tower can be effectively introduced to the market, alleviating the strain on pickup robots and facilitating efficient management of electronic components.

ACKNOWLEDGMENT

This work was supported by the Technology development program(S2958623) funded by Ministry of SMEs and Startups(MSS, Korea)

REFERENCES

[1] S.H. Park, K.H. Lee, J.S. Park, and Y.S. Shin. 2022. Deep Learning-based defect detection for sustainable smart manufacturing. MDPI Sustainability, Vol.14, No.5.

[2] S.H. Jang and J.P. Jeong. 2022. Design and Implementation of OpenCV-based Inventory Management System to build Small and Medium Enterprise Smart Factory. The Journal of The Institute of Internet, Broadcasting and Communication. 77-96.

[3] Zaramba, Wojciech, Ilya sutskever and Oriol Vinyals. 2015. Recurrent Neural Network Regularization. Neural and Evolutionary Computing. arXiv:1409.2329.

[4] Hochreiter. S. & Schmidhuber. J. 1997. LONG SHORT-TERM MEMORY. NEURAL COMPUTATION 9(8):1735-1780.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv:1412.3555

Author Index

Ahmad Faza	11
Akitoshi Hanazawa	111
Andrew Gahwera	76
Angelina Ervina Jeanette Egeten	45
Anuchit Jitpattanakul	155, 161, 167
Arit Thammano	51, 57
Arnando Harlianto	82
Arnel C. Fajardo	106
Aulia Kharisma Putri	11
Barasha Mali	133
Bellouqi Mohamed Amine	22
Bongjun Kim	41
Chanatkit Harnnuengnit	62
ChangSoo Moon	178
Deyu Zhang	127
Duangjai Jitkongchuen	51, 57, 143, 149
Fatin Hasnat Shakib	137
Hamidah Jantan	101
Husni Teja Sukmana	16
Irfan Fari Ramadhan	68
Isaac Mugume	76
Ismail Assayad	22
JaeHoon Chung	178
Jakkrapan Sreekajon	121
Jansen Wiratama	11, 45, 89
Jantawan Monchanuan	51
Johan Setiawan	82
Junho Jeong	41
Ka Ho Brian Chor	35
Kanteera Mekruksavanich	173

Author Index

Karikarn Chansiri	35
Khodijah Hulliyah	16
Krittika Kantawong	51, 57
Lilibeth P. Coronel	106
Mauridhi Hery Purnomo	93
Md Al-Imran	137
Monika Evelin Johan	11, 89
Muhammad Destamal Junas	16
MyeongSu Jeong	178
Natdanai Kamkhad	51
Natthayada Thamchaikul	173
Nongnapas Sutthipornmaneewat	173
Odongo Steven Eyobu	76
Panchit Longpradit	51, 57
Parachaya Muentabutra	173
Park Jimin	41
Pattharaporn Thongnim	121
Ponnipa Jantawong	155, 161, 167
Puthyrom Tep	62
Ratchakoon Pruengkarn	117
Reza Juliandri	89
Rizka Amalia Putri	16
Rudy Winarto	93
Ruji Medina	106
Sackdavong Mangkhaseum	111
Saepul Aripiyanto	16
Sajid Faysal Fahim	137
Sakkayaphop Pravesjit	51, 57
Sakorn Mekruksavanich	155, 161, 167
Saksit Sabaiporn	51

Author Index

Samuel Ady Sanjaya	11, 68, 89
Sanjida Simla	137
Santo Fernandi Wijaya	11, 45
Sarochar Khambuo	62
Sarwar Jahan	137
Sathien Hunta	57
Siti Ummi Masruroh	16
Steve Chan	28
Sunil Duwal	111
Supakpong Jinarat	117
Surapong Uttama	127
Thanapat Kangkachit	143, 149
Thanaphon Phukseng	121
Thongpan Supong	143
Tokiuddin Ahmed	137
Ummu Fatihah Binti Mohd Bahrin	101
Uthai Phuyued	149
Wiwik Anggraeni	93
Worasak Rueangsirarak	127
Xinyu Wei	35
Yogesh Bhattarai	111

IEEE IEEE THAILAND SECTION IEEE Signal Processing Society THAILAND CHAPTER สคทส. CITT BIG DATA INSTITUTE

IBDAP 2024

THE 5TH INTERNATIONAL CONFERENCE ON BIG DATA ANALYTICS AND PRACTICES (IBDAP2024)

BANGKOK, THAILAND
AUGUST 23 – 25, 2024

ORGANIZED BY
BIG DATA INSTITUTE (PUBLIC ORGANIZATION)
https://www.ibdap.org/